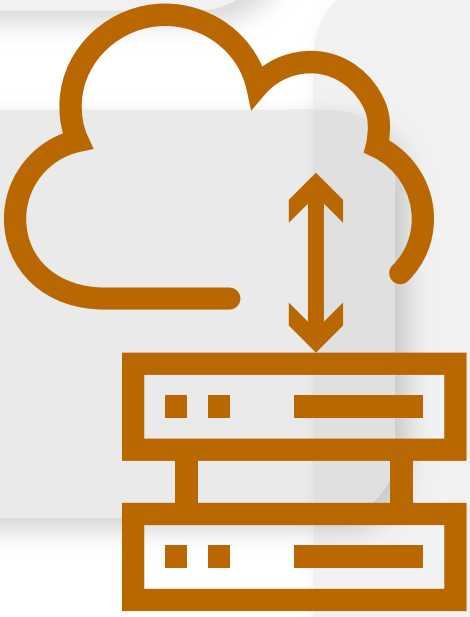
SAP PRESS

SAP PRESS E-Bites

Cloud Connector for SAP® Cloud Platform

How-to Guide



Morten Wittrock

Rheinwerk Publishing

Morten Wittrock

Cloud Connector for SAP® Cloud Platform: How-to Guide

Rheinwerk Publishing

SAP PRESS E-Bites

SAP PRESS E-Bites provide you with a high-quality response to your specific project need. If you're looking for detailed instructions on a specific task; or if you need to become familiar with a small, but crucial sub-component of an SAP product; or if you want to understand all the hype around product xyz: SAP PRESS E-Bites have you covered. Authored by the top professionals in the SAP universe, E-Bites provide the excellence you know from SAP PRESS, in a digestible electronic format, delivered (and consumed) in a fraction of the time!

Donna Leong-Cohen and Seng-Ping Gan
Integrating SAP SuccessFactors with SAP Cloud Platform Integration
ISBN 978-1-4932-1527-0 | $24.99 | 106 pages

Aron MacDonald
Integrating SAP HANA and Hadoop
ISBN 978-1-4932-1293-4 | $12.99 | 105 pages

Craig Cmehil
Hands On with SAP HANA Cloud Platform for IoT
ISBN 978-1-4932-1455-6 | $14.99 | 68 pages

The Author of this E-Bite

Morten Wittrock is an SAP technologist and alumnus of the SAP Mentor program. He has been working with SAP technologies since 2004, with a particular focus on integration. Morten was the founder and lead developer of the Detroubulator open source project for SAP Process Integration, and he is the organizer of the SAP Inside Track Copenhagen event.

What You'll Learn

Connect your on-premise systems to SAP Cloud Platform with Cloud Connector! In this E-Bite you'll learn how to install, configure, operate, and secure Cloud Connector. Through the E-Bite's exercises, you'll also get hands-on experience with Cloud Connector on your own machine.

1 Cloud Connector and the Hybrid Cloud Landscape

Before we begin working with Cloud Connector, let's take a step back and look at how Cloud Connector fits into the overall SAP cloud landscape. We'll discuss what hybrid cloud is and the connectivity challenge introduced by the hybrid cloud landscape. We'll then cover hybrid cloud applications on SAP Cloud Platform and how Cloud Connector addresses the connectivity challenge.

Note

The instructions in this E-Bite are based on Cloud Connector 2.10.1.

1.1 What Is Hybrid Cloud?

Hybrid cloud is a computing landscape that combines applications, data, and services from on-premise data centers, private clouds, and public clouds.

Note

A private cloud makes cloud computing capabilities available to a single customer. It can run in either the customer's own data center or in a data center operated by a vendor. However, unlike public cloud services, the resources of a private cloud are not shared.

In the more specific context of an existing SAP customer, hybrid cloud translates to a landscape where the customer runs a number of SAP systems either on-premise or in a private cloud and complements them with public cloud services and applications like SAP SuccessFactors (Figure 1).

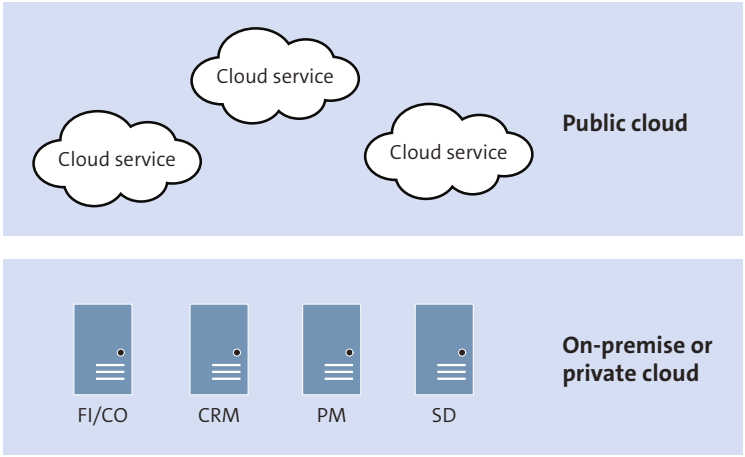


Figure 1 Hybrid Cloud in the Context of an On-Premise SAP Customer

For a customer with a large investment in on-premise SAP solutions, the hybrid cloud model makes a lot of sense. It lets the customer gain the benefits of cloud services, such as faster innovation cycles, scalability, and utility pricing, while keeping control of the core business systems in-house. Furthermore, the hybrid cloud model lets the customer adopt cloud services gradually and thus build both confidence in and experience with the cloud.

1.2 The Connectivity Challenge of Hybrid Cloud

In a hybrid cloud landscape, the on-premise and cloud applications and services *could* remain independent of each other. However, to realize the full benefit of the model, applications and services must be integrated across cloud and on-premise landscapes.

In the SAP world, examples of such integrations are SAP SuccessFactors synchronizing employee data with an on-premise SAP system, and an SAPUI5 application running on SAP Cloud Platform calling the OData API of an on-premise SAP S/4HANA system.

There is an obvious problem, though: your on-premise SAP systems are not—and should not be—reachable directly from the Internet. In fact, right

now your security department is working hard at making sure things stay that way. This is the connectivity challenge of hybrid cloud, which we must address.

1.3 Hybrid Cloud Applications on SAP Cloud Platform

SAP Cloud Platform is the platform for building hybrid cloud applications in SAP's cloud landscape. SAP Cloud Platform is SAP's platform-as-a-service (PaaS) product, offering a wide range of development tools and services in categories such as user experience, storage, integration, machine learning, mobile, Internet of Things, and many others.

Note

Platform-as-a-service (PaaS) is a cloud computing environment consisting of infrastructure and services that let customers build, run, and manage cloud applications and solutions. SAP Cloud Platform is a public cloud offering, but other vendors market private cloud PaaS products.

You can think of SAP Cloud Platform as the development stack for the SAP cloud world. SAP Cloud Platform is where you build new applications and solutions native to the cloud, extend existing SAP software solutions, and integrate the cloud and on-premise worlds.

And SAP Cloud Platform wouldn't be complete, of course, without an answer to the hybrid cloud connectivity challenge. That answer is Cloud Connector.

1.4 Cloud Connector's Approach to Connectivity

Cloud Connector is a software agent running on a host in the corporate network, in either the demilitarized zone (DMZ) or the internal network. Unlike a reverse proxy such as SAP Web Dispatcher, the Cloud Connector host is never connected to via the Internet. Instead, Cloud Connector establishes a

transport layer security (TLS)-encrypted tunnel to SAP Cloud Platform, which is used for all subsequent communications between the cloud and on-premise. This approach is known as *reverse invoke*.

Figure 2 shows Cloud Connector providing connectivity between SAP Cloud Platform and the internal network.

Note

In Figure 2, arrows represent network connections and point *away* from the system initiating the connection.

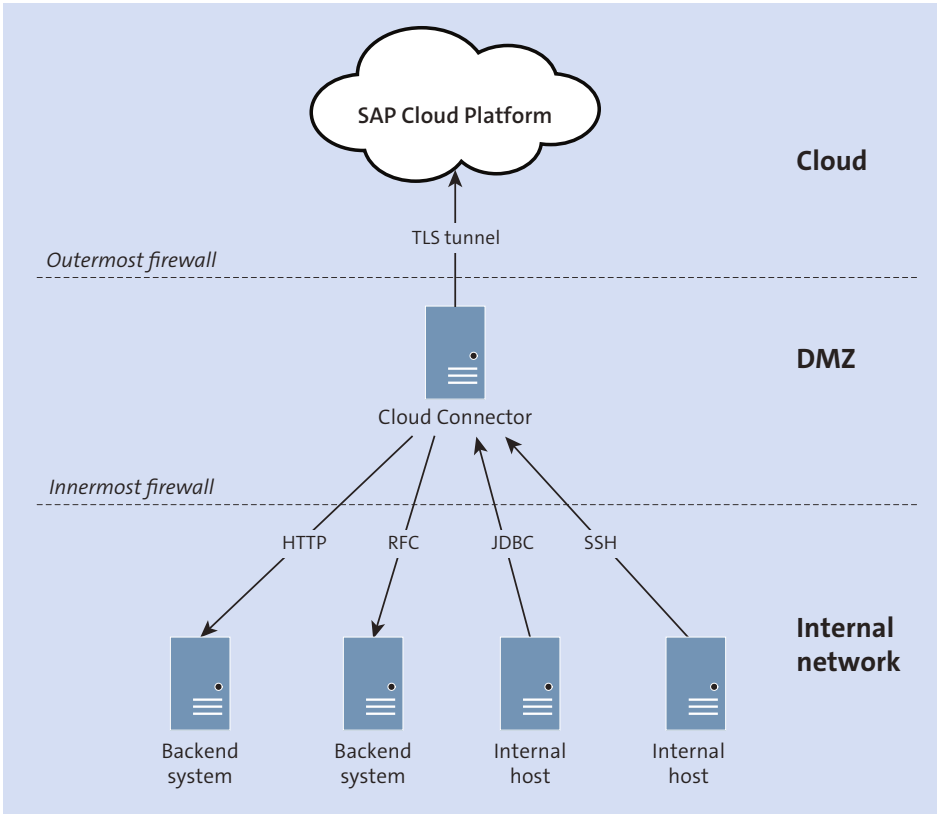


Figure 2 SAP Cloud Platform and the Internal Network Connected through Cloud Connector

Note

Figure 2 shows Cloud Connector running in the DMZ. However, it is also possible to run Cloud Connector in the internal network. This topic is discussed in Section 7.2.

Since no connections are made to the Cloud Connector host from the Internet, firewalls can block all inbound traffic to it. This shields the Cloud Connector host from denial-of-service attacks, intrusion attempts, and other nefarious activity.

When an SAP Cloud Platform application needs to send, for example, an HTTP request to a backend system, it doesn't communicate directly with that backend system. Instead, Cloud Connector receives the request through the TLS tunnel. If Cloud Connector's security mechanisms allow the request, Cloud Connector then communicates with the backend system on the application's behalf, receives the response, and returns it to the application through the TLS tunnel.

Cloud Connector can also be configured to provide access to cloud resources, such as virtual machines and SAP HANA databases running on SAP Cloud Platform, from on-premise hosts. Like the cloud to on-premise scenario, an on-premise host never communicates directly with a cloud resource. Instead, it communicates with Cloud Connector, which then communicates with the cloud resource through the TLS tunnel.

Note

Out of the box, Cloud Connector does not provide access to any backend or cloud resources. For a resource to be reachable, it must be explicitly configured in Cloud Connector.

2 Installing and Managing Cloud Connector

With the big picture in place, it's time to get practical and set up Cloud Connector on your machine. In this section, you will learn about Cloud Connector's system requirements, the two installation options available to you, and how to choose the one that's right for you. We cover how to install Cloud Connector and perform the initial setup. You will also learn how to start and stop Cloud Connector, how to upgrade a Cloud Connector installation, and how to uninstall the software.

2.1 Hardware Requirements

The hardware requirements of Cloud Connector are quite modest. At the time of writing, SAP recommends 4 GB of RAM, 20 GB of hard drive space, and a 2 GHz, dual-core x86-64 compatible CPU. This means that Cloud Connector runs well on most machines, from laptops and desktop PCs to physical and virtual servers. You can find the current hardware requirements in the **Hardware** section of the documentation's Prerequisites page (https://goo.gl/FJeAPH).

Note

To get an overview of the resources consumed by a Cloud Connector installation, go to the **Connector • Hardware Metrics Monitor** view.

2.2 Java Requirements

Cloud Connector is a Java application, and as such it requires Java installed on the machine it is running on. At the time of writing, Cloud Connector supports versions 7 and 8 of both Oracle's 64-bit JDK and SAP's own 64-bit

JVM. To get the latest Java compatibility information, please go to the documentation's Prerequisites page (https://goo.gl/FJeAPH) and scroll down to the **Supported JDKs** section.

Note

Even though Cloud Connector supports Java version 7, please be aware that the hands-on exercise in Section 4 requires a Java version 8 runtime.

2.3 Installation Options: Portable versus Installer

Cloud Connector offers two installation options: the portable version and the installer version. In the following sections, we cover the two options and discuss how to choose the best option for your circumstances.

Note

The installation options pertain *only* to how Cloud Connector is installed. The software that runs afterwards is the same in either case.

The Portable Version

The portable version is delivered as an archive, which you install simply by extracting it into a directory of your choice. Similarly, you uninstall it by deleting that directory. The portable version does not otherwise alter the system it is installed on. An additional benefit is that you can install the portable version without administrator privileges.

It has drawbacks, however. First, without administrator privileges you cannot make the portable version run continually and across system restarts. Furthermore, the portable version can be upgraded only to a limited extent, and the upgrade process is completely manual.

SAP does not recommend using the portable version in production scenarios.

Note

At the time of writing, the portable version is the only installation option available for the macOS operating system.

The Installer Version

The installer version is delivered as a Windows Installer file for Microsoft Windows (an *.msi* file) or an RPM package for Linux (an *.rpm* file). It is not available for the macOS operating system.

In contrast to the portable version, the installer version runs continually in the background, and it will start up automatically after a system restart. It runs as a service on Microsoft Windows and as a daemon on Linux. The installation requires administrator privileges, however, which is the other major difference between the portable version and the installer version. Additionally, the installer version can be automatically upgraded to a newer release of Cloud Connector.

The installer version is the option SAP recommends for production scenarios.

Which Version Should You Choose?

If you are running the macOS operating system, or if you don't have administrator privileges on the installation machine, you have only one choice: the portable version.

As for production scenarios, the choice is also simple: use the installer version, as recommended by SAP.

In all other cases, both options are available to you. If you are setting up Cloud Connector locally for learning purposes, going with the installer version makes a lot of sense, since that's what you will be running in production. However, if you prefer an installation that is self-contained and easy to remove, go with the portable version instead.

2.4 Hands-on: Installing Cloud Connector

In this hands-on exercise, you will install Cloud Connector on your local machine. There are three steps to this exercise. In the first step, you download the latest version of Cloud Connector to your machine. In the next step, you perform the actual installation, and in the final step, you log in and complete the initial setup of Cloud Connector.

Note

If you are installing Cloud Connector in a corporate network, please make sure that you do so with the blessing of the security department. For more information about Cloud Connector security, please see Section 7.

Download the Latest Version of Cloud Connector

To download the latest version of Cloud Connector, go to the SAP Development Tools page (https://tools.hana.ondemand.com#cloud). Scroll down to the **Cloud Connector** section, where you will find the download links. Click the download link for your operating system and installation option of choice. When you have agreed to the license agreement, the download starts.

Note

If you want to make sure that your operating system version is compatible with your Cloud Connector version, go to the documentation's Prerequisites page (https://goo.gl/FJeAPH) and scroll down to the **Product Availability Matrix** section.

The next three sections cover installation of the portable version on all three operating systems, the installer version on Microsoft Windows and the installer version on Linux. Please continue to the section that covers your operating system and chosen installation option.

Install the Portable Version

The portable version is delivered as a *.zip* archive on Microsoft Windows, and a *.tar.gz* archive on Linux and macOS. To install it, simply extract the archive's directories and files to a directory of your choosing.

All three operating systems offer a GUI tool for extracting archives. On macOS and Linux you can extract the files from a shell with the following command:

```
tar xzf <the downloaded .tar.gz file>
```

The scripts that start the portable version of Cloud Connector must be able to execute the `java` command. For that to work, either your Java installation's *bin* directory must be added to the `PATH` environment variable, or the `JAVA_HOME` environment variable must be set, pointing to your Java installation directory.

Install the Installer Version on Microsoft Windows

The Windows installer version of Cloud Connector has one additional requirement: You need to install the Visual C++ Redistributable Package for Visual Studio 2013. The file, *vcredist_x64.exe*, can be downloaded from Microsoft's website (https://goo.gl/i99oa3).

Note

The Visual C++ Redistributable Package for Visual Studio 2013 might already have been installed on your system by another software package. In that case, you don't have to do anything.

The installer version on Microsoft Windows is delivered as a Windows Installer file. To launch the installation wizard, double-click the downloaded file. The wizard's first step, shown in Figure 3, welcomes you and informs you that you are about to install Cloud Connector. In the following sections, we walk you through the wizard's remaining steps.

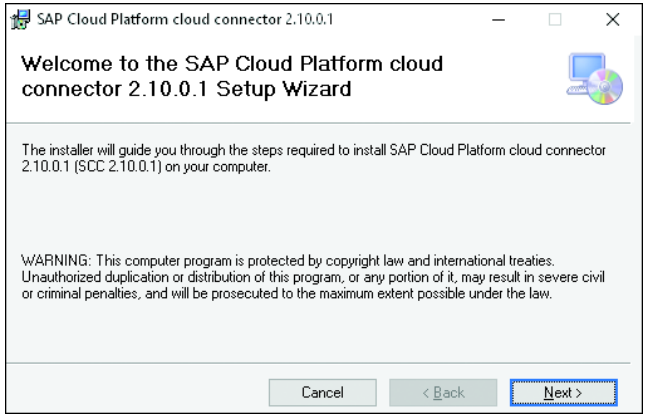


Figure 3 The First Step of the Microsoft Windows Installation Wizard

Click the **Next** button to proceed to the next step.

Choose Installation Folder

In the wizard's second step, shown in Figure 4, you choose where to install Cloud Connector.

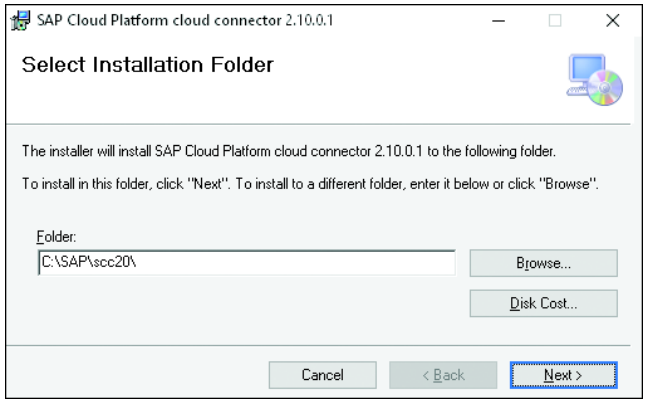


Figure 4 Choosing an Installation Directory in the Microsoft Windows Installation Wizard

Click the **Browse...** button and choose the directory you want to install Cloud Connector in, or accept the default installation directory. Then click the **Next** button to proceed to step three.

Choose Port Number

In the installation wizard's third step, shown in Figure 5, you choose the port number that the Cloud Connector UI should be made available on.

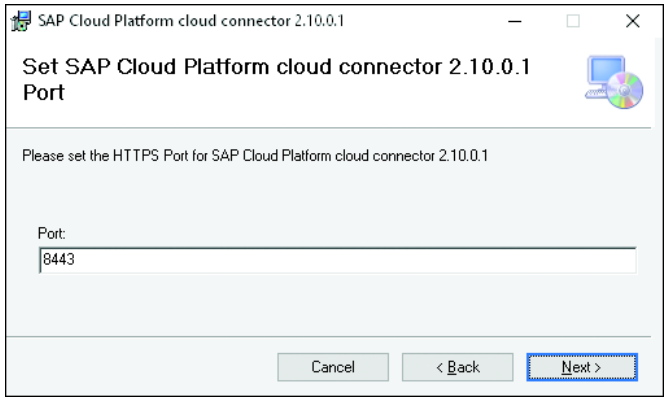


Figure 5 Choosing a Port Number in the Microsoft Windows Installation Wizard

Enter a port number of your choosing or accept the default port number 8443. Then click the **Next** button to proceed to the next step of the wizard.

Select Java Installation

In the fourth step of the wizard, shown in Figure 6, you must guide the wizard to the location of your Java installation.

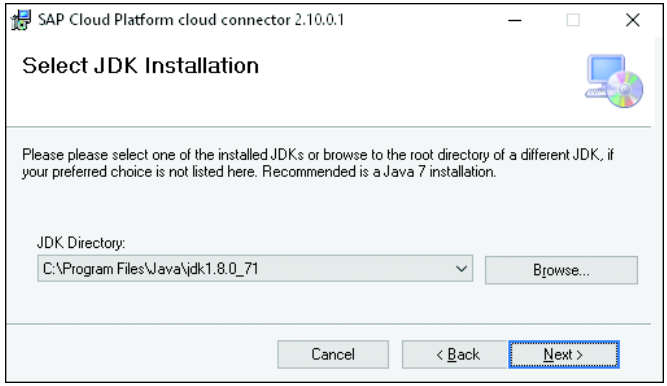


Figure 6 Selecting a Java Installation in the Microsoft Windows Installation Wizard

The **JDK Directory** drop-down list contains the Java installations that the installer knows about. If the Java installation you want to use is not in the list, click the **Browse...** button and choose the Java installation directory. Otherwise, select your Java installation from the drop-down list. Then click the **Next** button.

Start Service after Installation

In the installation wizard's fifth step, shown in Figure 7, you indicate whether you want to start the Cloud Connector Windows service when the installation is complete.

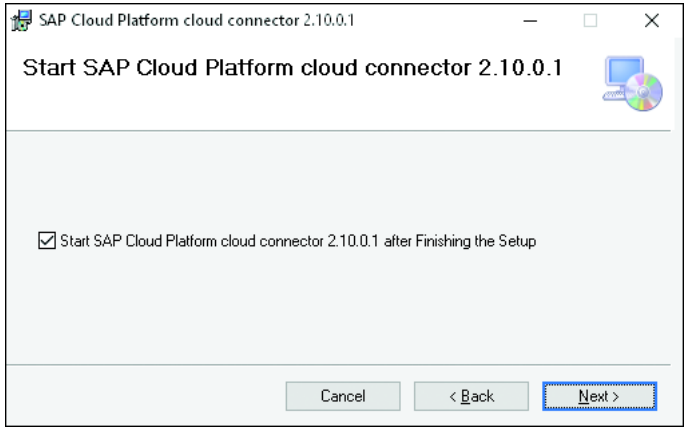


Figure 7 Choosing Whether to Start Cloud Connector after Installation in the Microsoft Windows Installation Wizard

If you want to manually start the service later, uncheck the check box. Otherwise, leave it checked. Then click the **Next** button to proceed to the next step.

Confirm Installation

In the sixth step of the wizard, shown in Figure 8, you are prompted to confirm that you want to go ahead with the installation.

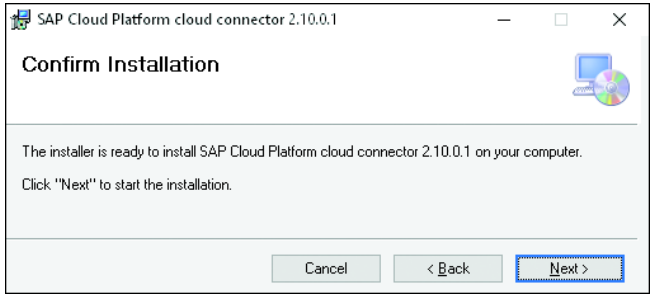


Figure 8 Confirming That You Want to Start the Installation in the Microsoft Windows Installation Wizard

If you want to change any of the information you have entered, you can navigate backward through the wizard's steps by clicking the **Back** button. Otherwise, click the **Next** button to start the installation.

The Cloud Connector installation requires administrative privileges, so at this point, Microsoft Windows's User Account Control mechanism prompts you to allow elevating the installer's privileges.

Installation Summary

When the installation is complete, you see the summary shown in Figure 9.

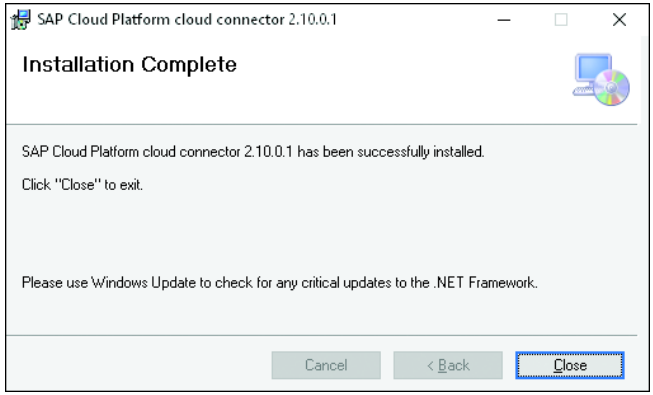


Figure 9 The Microsoft Windows Installation Wizard Has Completed the Installation

Click the **Close** button to exit the installation wizard.

Note

Cloud Connector does not use any components of the .NET framework, even though this last step of the installation wizard seems to indicate otherwise.

Install the Installer Version on Linux

The Linux installer version of Cloud Connector is delivered as a *.zip* archive containing an RPM package. To start the installation, extract the RPM file using either your desktop environment's built-in archive tool or the following command from the command line:

```
unzip <the downloaded .zip archive>
```

Next, use the `rpm` command to install the package by executing the following command:

```
rpm -i <the extracted .rpm file>
```

The command requires root privileges, so you must either run it as the root user or prefix it with the `sudo` command as follows:

```
sudo rpm -i <the extracted .rpm file>
```

When the `rpm` command completes, Cloud Connector has been installed and the daemon is running in the background.

Log In and Perform Initial Setup

With Cloud Connector installed and started, you are ready to log in for the first time. To do so, go to https://localhost:8443/ in your browser (if you chose a different port number during installation, alter the URL accordingly). If everything is running correctly, you are presented with the login screen shown in Figure 10.

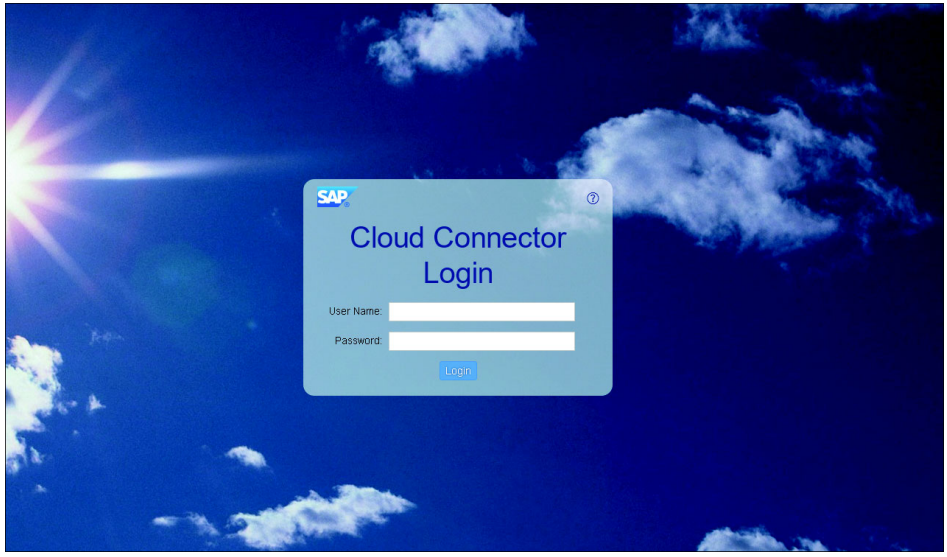


Figure 10 The Cloud Connector Login Screen

To log in for the first time, enter “Administrator” into the **User Name** field and the default password “manage” into the **Password** field. Click the **Login** button, and you will be taken to the **Initial Setup** screen, shown in Figure 11.

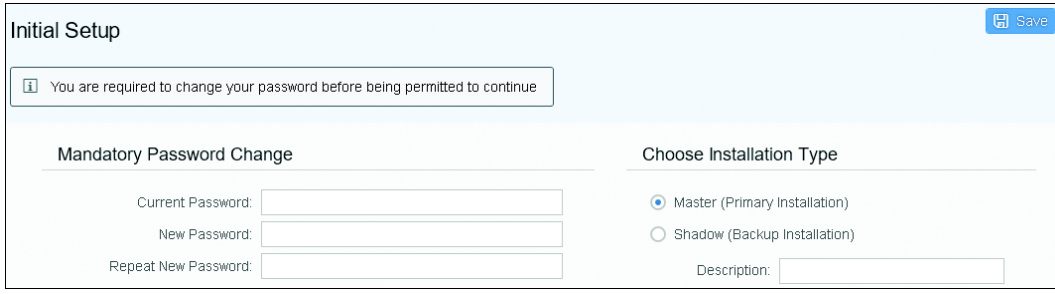


Figure 11 The Initial Setup Screen

When logging in for the first time, it is mandatory to set a new password for the administrator user. To do so, enter the default password “manage” into the **Current Password** field and a new password of your choosing into the **New Password** and **Repeat New Password** fields.

The **Choose Installation Type** setting determines whether this is the primary installation or the backup installation in a high availability setup. For this hands-on exercise, leave the setting at **Master (Primary Installation)**. To learn more about Cloud Connector's high availability setup, please see Section 6.6.

Click the **Save** button to store the settings and finish the initial setup.

2.5 Starting and Stopping Cloud Connector

How to start and stop Cloud Connector depends on both your chosen installation option and your operating system. The following sections cover how to start and stop the portable version on Microsoft Windows, Linux, and macOS and the installer version on Microsoft Windows and Linux.

Portable Version

To start the portable version of Cloud Connector, you need to run a script, either by double-clicking it or starting it from the command line. The script is located in the root of the installation directory. On Microsoft Windows, it is a batch file called *go.bat*. On both Linux and macOS, it is a shell script called *go.sh*.

To stop the portable version of Cloud Connector, simply exit the running script by pressing Ctrl+C or closing the terminal window that the script is running in.

Installer Version on Microsoft Windows

On Microsoft Windows, the installer version of Cloud Connector runs as a service in the background. Services are managed in Microsoft Management Console using the Services snap-in. You can launch the Services snap-in from the Control Panel by searching for "services" in the search box and clicking on **View local services** in the search results. Alternatively, open the

Run dialog by pressing Windows+R, enter “services.msc” into the **Open** field, and click **OK** or press Enter.

Once the Services snap-in has launched, locate the Cloud Connector service in the list of services and select it. You can now stop, start, pause, and restart the service using the tools highlighted in Figure 12.

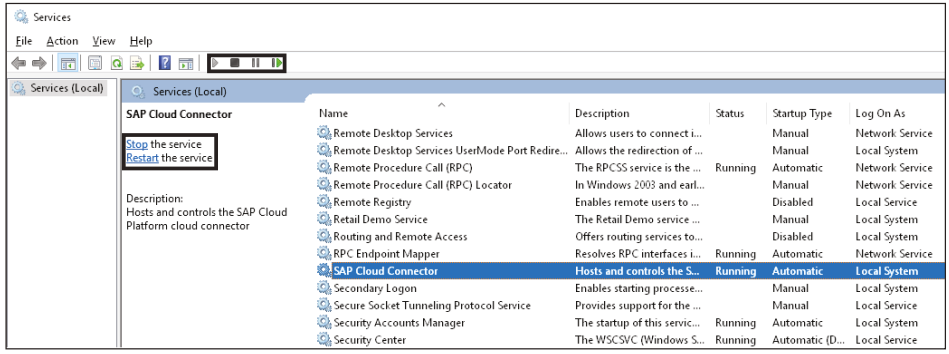


Figure 12 Microsoft Windows 10 Services Snap-in Tools for Managing the Cloud Connector Service

Alternatively, you can use the two desktop shortcuts that the Cloud Connector installer creates to start and stop the service. The shortcuts are shown in Figure 13.

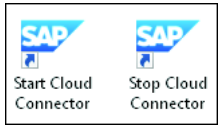


Figure 13 Microsoft Windows Desktop Shortcuts Created by the Cloud Connector Installer

Installer Version on Linux

On Linux, the installer version of Cloud Connector runs as a daemon in the background. Daemons can be started, stopped, and restarted using the `service` command. To do so, run the following command from the command line:

```
service scc_daemon start|stop|restart
```

The command requires root privileges, so either execute it as the root user or prefix it with the `sudo` command like this:

```
sudo service scc_daemon start|stop|restart
```

Restarting from the UI

In addition to the methods described in the previous sections, you can also trigger a restart of Cloud Connector from within the UI. To do so, click the **Restart** button, which has been highlighted in Figure 14. This method works across installation options and operating systems.

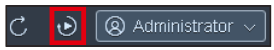


Figure 14 The Restart Button in the Cloud Connector UI

2.6 Uninstalling Cloud Connector

How to remove Cloud Connector also depends on your operating system and your chosen installation option. In the following sections, you will learn how to uninstall the portable version on Microsoft Windows, Linux, and macOS and the installer version on Microsoft Windows and Linux.

Portable Version

Uninstalling the portable version of Cloud Connector on Microsoft Windows, Linux, and macOS is as easy as installing it: remove the installation directory, and you're done. Keep in mind, though, that this also removes the configuration files, so if you would like to keep a copy of them, make a backup beforehand. For more information on backing up your configuration, please see Section 6.3.

Installer Version on Microsoft Windows

To remove the installer version on Microsoft Windows, go to the Control Panel. In the **Programs** section, select **Uninstall a program**. Use the search

box to search for “cloud connector”, select Cloud Connector in the search results, and click the **Uninstall** button, as shown in Figure 15.

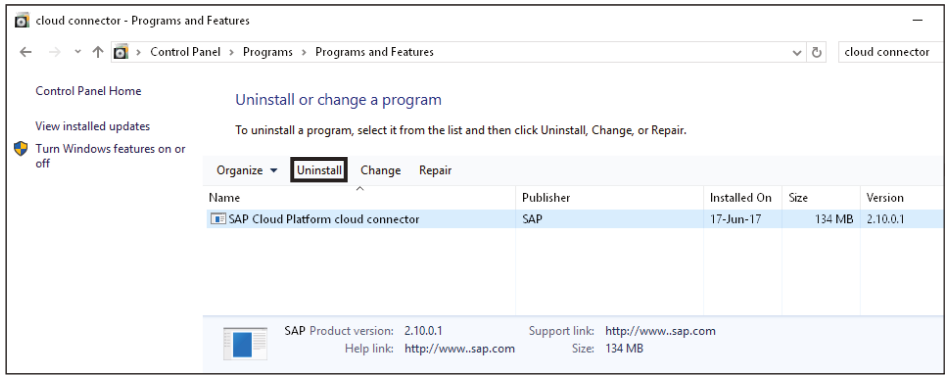


Figure 15 Uninstalling Cloud Connector on Microsoft Windows 10

During the uninstall procedure, you have the option of backing up Cloud Connector’s configuration files, as shown in Figure 16.

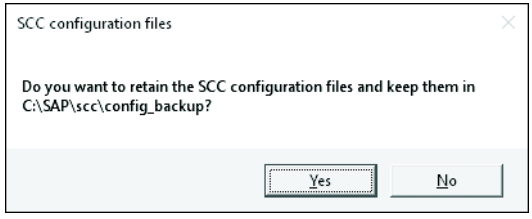


Figure 16 Backing Up Cloud Connector’s Configuration Files during the Microsoft Windows Uninstall Procedure

If you choose not to perform the backup, the current configuration will be lost.

Installer Version on Linux

The installer version on Linux is removed using the `rpm` package manager tool, which we also used to install Cloud Connector. To uninstall, execute this command from the command line:

```
rpm -e com.sap.scc-ui
```

The command requires root privileges, so either run it as the root user or prefix it with the `sudo` command as follows:

```
sudo rpm -e com.sap.scc-ui
```

Warning

The Linux uninstall procedure removes the Cloud Connector configuration files without warning. If you want to keep a copy of them, make sure to back them up *before* performing the uninstall. For more information on backing up the configuration files, please see Section 6.3.

2.7 Upgrading an Existing Installation

The installer version of Cloud Connector can be upgraded to a newer release, and the portable version can be upgraded to a limited extent. In the following sections, you will learn how to perform an upgrade of the installer version on both Microsoft Windows and Linux, as well as an upgrade of the portable version.

Upgrading a Cloud Connector installation takes it offline for a brief period of time, but with the high availability setup in place, this downtime can be minimized. For more information on how to do this, please see Section 6.7.

Also, a newer Cloud Connector version might bring with it updated UI elements, so remember to refresh your browser cache after an upgrade. This ensures that you are, in fact, seeing the updated UI. In most browsers, Ctrl + F5 reloads the page and forces a cache refresh.

Portable Version

There is no automatic upgrade procedure for the portable version of Cloud Connector. You can, however, with certain restrictions, transfer your current configuration to an installation of a newer version.

To do this, back up your current configuration, install the newer portable version, and restore the backed-up configuration files to the newer version's installation directory. The process of backing up and restoring the configuration files is described in Section 6.3.

Be aware, though, that SAP recommends against restoring backed-up configuration files across major and minor versions of Cloud Connector. For example, you should not restore version 2.9.x configuration files to version 2.10.y. Unfortunately, this significantly limits the viability of upgrading the portable version.

Installer Version on Microsoft Windows

Upgrading the installer version of Cloud Connector on Microsoft Windows is a two-step process. First, you uninstall the old version, making sure to retain a backup of its configuration files (the Microsoft Windows uninstall procedure was covered in Section 2.6). Second, you install the new version in the same directory as the old version. When the Microsoft Windows installer detects the presence of the backed-up configuration files, it automatically transfers the old version's configuration to the new version.

Installer Version on Linux

To upgrade the installer version on Linux, you use the same `rpm` package manager tool that you use to install and uninstall Cloud Connector. To perform the upgrade, download the newer version and execute the following command from the command line:

```
rpm -U <the newer version's .rpm file>
```

As is the case when installing and uninstalling Cloud Connector, this command requires root privileges, so either run it as the root user or prefix it with the `sudo` command as follows:

```
sudo rpm -U <the newer version's .rpm file>
```

3 Connecting to SAP Cloud Platform

In Section 2, you installed Cloud Connector and completed the initial setup. At this point, however, Cloud Connector isn't really useful. The first step toward remedying that is to connect the newly installed Cloud Connector to your SAP Cloud Platform subaccount, which we do in this section.

We also cover the network access required by Cloud Connector and how to disconnect Cloud Connector from an SAP Cloud Platform subaccount.

In some customer landscapes, on-premise systems are distributed across more than one geographical location. To support this scenario, you need to connect one Cloud Connector per location to the same SAP Cloud Platform subaccount, and in this section, we discuss how to do so.

3.1 Network Requirements

A machine running Cloud Connector must be able to connect to a small number of SAP Cloud Platform hosts, in order to establish the TLS tunnel. This means either connecting directly through the corporate firewall or going through an HTTPS proxy. This section covers the details of connecting directly, and the HTTPS proxy option is covered in Section 3.3.

On your laptop or desktop machine at home, connecting to hosts on the Internet is not a problem. But in a production environment, outbound connections may not be allowed by the firewall.

If Cloud Connector is unable to connect to SAP Cloud Platform, the *ljs_trace.log* trace file will contain a "Permission denied" error message (see Section 6.2 for more information about logs and traces). In that case, you need to get in touch with your firewall administrator about allowing outbound connections from Cloud Connector to SAP Cloud Platform.

Note

Which firewalls to update, depends on where Cloud Connector is running. For more information on this topic, please see Section 7.2.

The specific hosts to which connections must be allowed depend on the region your SAP Cloud Platform global account belongs to. The host names and IP addresses for the European trial region are listed in Table 1. You can find the full and up to date list for all SAP Cloud Platform regions in the **Network** section of the documentation's Prerequisites page (https://goo.gl/FJeAPH).

Host name	IP address
connectivitynotification.hanatrial.ondemand.com	155.56.219.26
connectivitycertsigning.hanatrial.ondemand.com	155.56.219.22
connectivitytunnel.hanatrial.ondemand.com	155.56.219.27

Table 1 Host Names and IP Addresses That Cloud Connector Needs Access To in the European Trial Region

For all the listed SAP Cloud Platform hosts, the port number to allow connections to is 443 (the default HTTPS port).

3.2 Hands-on: Connecting to an SAP Cloud Platform Trial Subaccount

The first step toward providing access to cloud and on-premise resources through Cloud Connector is to connect it to an SAP Cloud Platform subaccount.

In this hands-on exercise, you will connect the Cloud Connector instance that you installed in Section 2 to the SAP Cloud Platform trial region. If you don't have a trial account yet, please register for one by visiting the SAP Cloud Platform website (https://cloudplatform.sap.com/) and clicking the **Try it for Free** button.

The first thing you see after completing Cloud Connector's initial setup is the **Define Subaccount** screen, shown in Figure 17.

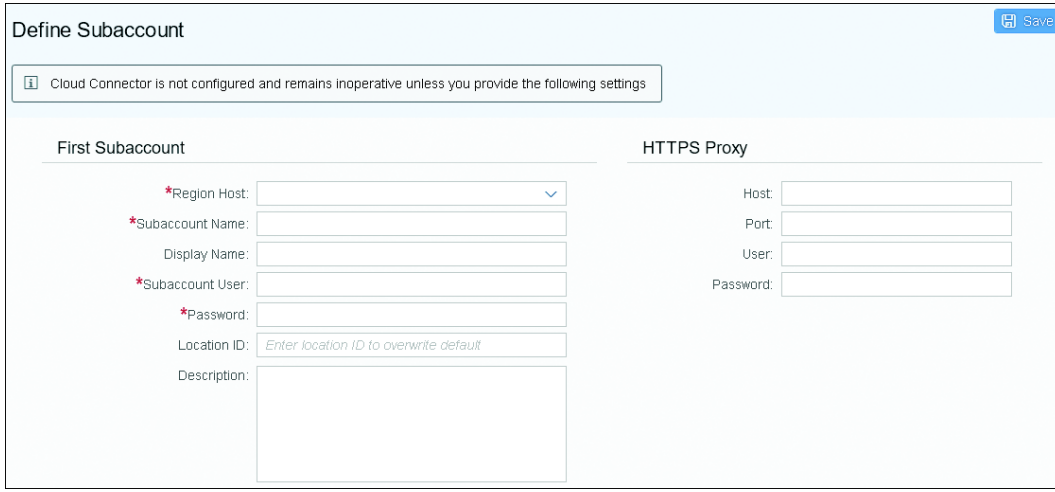


Figure 17 Adding the First SAP Cloud Platform Subaccount

From the **Region Host** drop-down list, choose the appropriate host for the region that your SAP Cloud Platform global account belongs to. Since we are connecting to the trial region, choose **hanatrial.ondemand.com**.

The **Subaccount Name** field must be filled out with the name of your SAP Cloud Platform subaccount. If you are in doubt as to what that is, navigate to the subaccount dashboard in the SAP Cloud Platform cockpit by clicking the subaccount in the cockpit's account navigation bar, as highlighted in Figure 18.

Figure 18 The Subaccount in the Account Navigation Bar

Note

In the trial region, there is no global account, so the navigation bar shows only the trial region and the subaccount. For enterprise accounts (i.e., non-trial accounts), the navigation bar shows the region, the global account, and the subaccount.

Scroll to the **Subaccount Information** section at the bottom of the page, where you will find the subaccount name, as highlighted in Figure 19. In the trial region, the subaccount name is always your trial user name with "trial" appended to it.

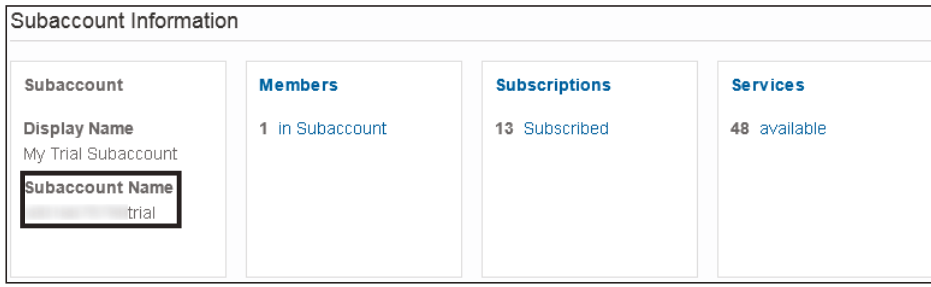


Figure 19 Locating Your Subaccount Name on the Subaccount Dashboard

In the **Display Name** field, you can optionally enter a descriptive name for the subaccount. The name you enter here is used in the Cloud Connector UI in place of the subaccount name.

The SAP Cloud Platform subaccount user that you wish to establish the connection with must be entered into the **Subaccount User** field. Enter the user's password in the **Password** field. For more information on the user you connect to SAP Cloud Platform with please see Section 3.5. For this hands-on exercise, enter your trial user (i.e., the user referenced in the SAP Cloud Platform trial welcome email) and its password.

The value entered into the **Location ID** field is used to differentiate between multiple Cloud Connector instances connected to the same SAP Cloud Platform subaccount. This scenario is covered in Section 3.6. Leave the field blank for this hands-on exercise.

You can optionally provide a description of the subaccount in the **Description** field.

Filling out the **HTTPS Proxy** section is only required if you need to go through a proxy to reach the Internet from the Cloud Connector host. For more information on this scenario, please see Section 3.3.

Having filled out the fields as described, click the **Save** button. Cloud Connector establishes the connection to the SAP Cloud Platform subaccount, and you will be taken to the **Connector** view. The newly added subaccount is now listed in the **Subaccount Dashboard** section of the page.

The subaccount's status icon ◇ indicates that the subaccount is connected but that no resources are available to it yet. The reason for this is simply that we haven't added any resources yet. We do so in Section 4.

To confirm that the tunnel has been established, log on to the SAP Cloud Platform cockpit of your trial subaccount and navigate to the **Connectivity • Cloud Connectors** view. There you will find the details of the newly connected Cloud Connector, as shown in Figure 20.

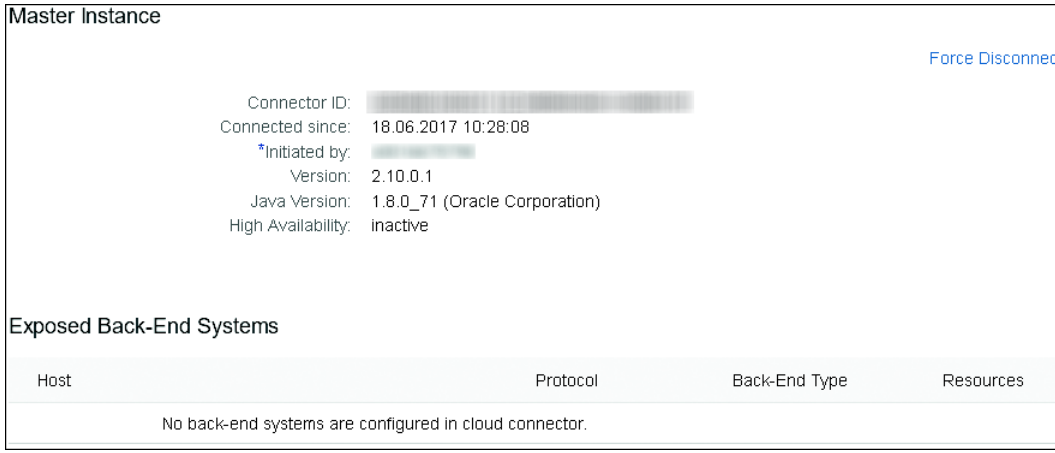


Figure 20 The Newly Connected Cloud Connector as It Appears in the SAP Cloud Platform Cockpit

3.3 Connecting Through an HTTPS Proxy

As we discussed in Section 3.1, Cloud Connector needs to be able to connect to a small number of SAP Cloud Platform hosts over the Internet to function. It can either connect directly through the corporate firewall or go through an HTTPS proxy server. In the latter case, you need to add the

details of the proxy in the **Define Subaccount** screen's **HTTPS Proxy** section, shown in Figure 21. If you don't have the required information, your network administrator can provide you with it.

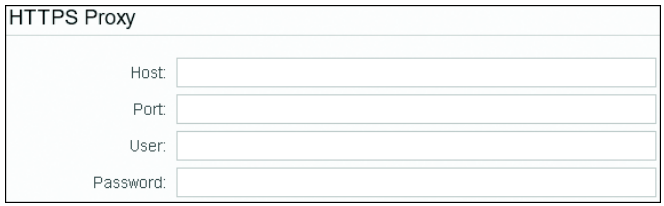


Figure 21 Adding an HTTPS Proxy

Enter the host name and port number of the proxy in the **Host** and **Port** fields, respectively. These two fields are mandatory. If the proxy requires authentication, enter the username into the **User** field and the password into the **Password** field.

If you need to add an HTTPS proxy later, or edit an already configured proxy, you can do so by going to the **CLOUD** tab of the **Connector • Configuration** view and clicking the **Edit** icon in the **HTTPS Proxy** section.

3.4 Connecting to Multiple SAP Cloud Platform Subaccounts

Should you need to, it is possible to connect a single Cloud Connector instance to multiple SAP Cloud Platform subaccounts. This has been an option since of Cloud Connector version 2.2.0.

As we shall see in Section 4, backend systems and resources on those systems, such as OData services and function modules, are added *per subaccount*. This means that the backend systems and resources that you add to one subaccount will not be available to another subaccount, unless you actively add them to that subaccount as well.

It is also possible to connect multiple Cloud Connector instances to a single SAP Cloud Platform subaccount. This is covered in Section 3.6. There is, in

other words, a many-to-many relationship between Cloud Connector instances and SAP Cloud Platform subaccounts.

Note

Be aware that SAP advises against connecting the same Cloud Connector instance to both development, test, and production SAP Cloud Platform subaccounts. The recommended approach is to have a separate Cloud Connector instance for production and at least one for development and testing.

3.5 A Closer Look at the Initiating User

The user account that you use to connect to SAP Cloud Platform must have the `Cloud Connector Admin` role in its subaccount. You can assign this role to the user in the **Members** view of the SAP Cloud Platform cockpit. If the user does not have the required role, adding the subaccount will fail, and an "SCC handshake failed" error message will appear in the *ljs_trace.log* trace file.

However, once the TLS tunnel from Cloud Connector to SAP Cloud Platform has been established, the user no longer needs the role. In fact, at that point the *user itself* is no longer needed. The user you provide is only needed to initiate the tunnel. If it is later removed from the subaccount, the tunnel will still function.

The user who initiated the connection is displayed in the list of connected Cloud Connector instances in the **Connectivity • Cloud Connectors** view in the SAP Cloud Platform cockpit, as highlighted in Figure 22.

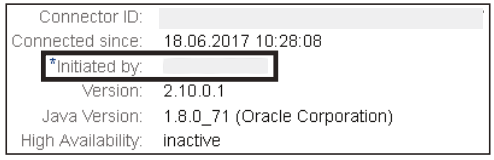


Figure 22 The User Who Initiated the Connection to SAP Cloud Platform

Note

In Section 3.2 you connected to the trial region with your trial user. You might have noticed that you didn't have to assign your trial user the `Cloud Connector Admin` role. That is because a trial user is automatically assigned the `Administrator` role, and that role also enables a user to initiate a connection from Cloud Connector to an SAP Cloud Platform subaccount. However, since the `Administrator` role grants many other privileges, you should never assign it to a user solely for the purpose of connecting a Cloud Connector instance to SAP Cloud Platform with that user. For that purpose, always use the much more specific `Cloud Connector Admin` role.

3.6 Connecting Multiple Cloud Connector Instances to the Same Subaccount

In some cases, you might need to connect more than one Cloud Connector instance to the same SAP Cloud Platform subaccount. For example, a company might have backend systems running in more than one geographical location. In earlier releases of Cloud Connector this was not possible, but version 2.9.0 added support for this scenario.

When you connected to the SAP Cloud Platform trial subaccount in Section 3.2, the **Location ID** field was left blank. This is the default value, but when more than one Cloud Connector instance is connected to the same SAP Cloud Platform subaccount, the location ID is used to differentiate between them. This means that at most one of them can have the default location ID; the rest must be assigned a unique value.

If you attempt to connect an additional Cloud Connector instance to an SAP Cloud Platform subaccount using a duplicate location ID (default or otherwise), the connection will fail, with the error message: "Unable to connect: A different cloud connector instance is already connected to this subaccount."

Note

Incidentally, this error message seems to have been repurposed. It *used* to indicate that a different Cloud Connector instance was already connected to the subaccount. That is no longer an error condition, however. Now it means that a different Cloud Connector instance is already connected using the same location ID.

Apart from assigning a location ID, connecting additional Cloud Connector instances to the same SAP Cloud Platform subaccount is identical to connecting the first one. Figure 23 shows the **Connectivity • Cloud Connectors** view in the SAP Cloud Platform cockpit, with two Cloud Connector instances connected to the subaccount. Note that the two location IDs are displayed.

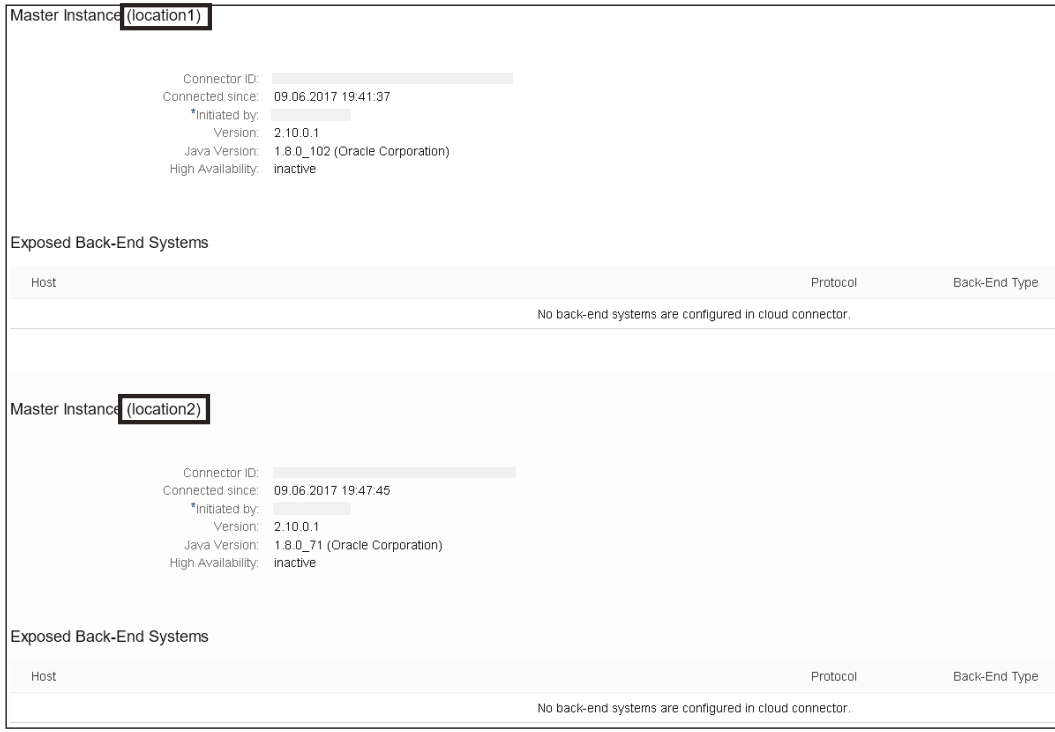


Figure 23 Two Cloud Connector Instances Connected to One Subaccount

Destinations and the Location ID

An SAP Cloud Platform destination wraps the details of a connection to a remote system. Each destination has a symbolic name that you can reference when you want to reach that particular system. Destinations, in other words, provide a level of indirection between the application and the connection information. If you are familiar with the on-premise SAP world, SAP Cloud Platform destinations are similar in concept to the RFC destinations of the ABAP stack.

If you have multiple Cloud Connector instances connected to your SAP Cloud Platform subaccount, you need to provide the location ID when you create a destination that should be routed through one of those Cloud Connector instances.

The location ID is supported directly in the **Connectivity • Destinations** view in the SAP Cloud Platform cockpit. Destinations of type RFC are implicitly connections to on-premise systems, so for that protocol, the **Location ID** field is always shown. HTTP and LDAP destinations, however, can point to both cloud and on-premise systems.

When you create a new HTTP or LDAP destination and set the value of the **Proxy Type** field to **OnPremise** (indicating that this destination should be routed through a Cloud Connector instance), the **Location ID** field is displayed, as shown in Figure 24.

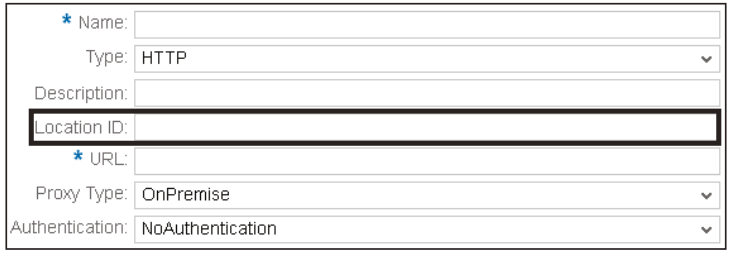


Figure 24 Adding the Location ID to an SAP Cloud Platform HTTP Destination

Fill out the field with the location ID of the Cloud Connector instance that this destination should be routed through, or leave it blank if the Cloud Connector instance in question has the default location ID.

3.7 Disconnecting Cloud Connector

When you disconnect Cloud Connector from an SAP Cloud Platform subaccount, all backend systems and resources added to the subaccount become unreachable from within that subaccount. However, all configuration remains in place in Cloud Connector. This means that when Cloud Connector is reconnected to the subaccount, all the configured resources become available once more.

Disconnecting from a particular subaccount is done from the **Subaccount Dashboard** section of the **Connector** view. Click the **Disconnect this subaccount** icon next to the subaccount that you want to disconnect from. You can achieve the same result by clicking the **Disconnect** button on the subaccount detail page. If you subsequently go to the **Connectivity • Cloud Connectors** view in the SAP Cloud Platform cockpit, you will find that the disconnected Cloud Connector instance is no longer listed there.

Should you find yourself in a situation in which the Cloud Connector instance is inaccessible, it is possible to disconnect it from within the SAP Cloud Platform cockpit. To do so, go to the **Connectivity • Cloud Connectors** view and click the **Force Disconnect** link next to the Cloud Connector instance that you wish to disconnect.

The ability to disconnect a Cloud Connector instance from within the SAP Cloud Platform cockpit was introduced in version 2.8. Earlier versions can be disconnected only from the Cloud Connector side, not from the cloud.

Note

SAP recommends using the remote disconnect option only if it is not possible to disconnect from within the Cloud Connector UI.

4 Configuring Cloud to On-Premise Access

With Cloud Connector installed and connected to SAP Cloud Platform, you are ready to learn how to configure it. The two scenarios supported by Cloud Connector are providing access to on-premise resources from the cloud and providing access to cloud resources from on-premise. This section covers the former scenario, while the latter is covered in Section 5.

We discuss the cloud to on-premise scenario in more detail and cover the protocols supported by Cloud Connector. You will then learn how to add an on-premise system to Cloud Connector and how to add on-premise resources, thereby making them available to SAP Cloud Platform applications and services.

We will cover how to map cookie domains, how to configure additional security for the RFC protocol, and how to set up Cloud Connector for RFC with Secure Network Communications. Then, you will set up a REST service on your own machine and call that service from an SAP Cloud Platform application. Finally, we will discuss how to access TCP resources via Cloud Connector.

4.1 Cloud to On-Premise Overview

The cloud to on-premise scenario is all about making your on-premise resources, such as ABAP function modules and SOAP, OData, or REST services, available to SAP Cloud Platform applications and services in a secure fashion. Some examples of this scenario are building extensions for an on-premise SAP S/4HANA system on SAP Cloud Platform, integrating SAP SuccessFactors with an on-premise SAP system, and building SAP Cloud Platform applications based on APIs exposed by on-premise systems.

The security of the cloud to on-premise scenario revolves around whitelisting of on-premise systems and resources. What that means is that Cloud Connector will never provide access to anything that you do not explicitly

allow. For more information on Cloud Connector security in general, please see Section 7.

4.2 Supported Protocols

The protocols supported by Cloud Connector determine the kinds of resources you can make available to SAP Cloud Platform applications and services. When the first version of Cloud Connector was released, HTTP was the only supported protocol, but more protocols have since been added. The following sections give you an overview of the currently supported protocols.

HTTP

The HTTP protocol has been supported since the first release of Cloud Connector, along with its encrypted version, HTTPS. It is the underlying protocol of the REST, OData, and SOAP services exposed by your on-premise systems.

RFC

RFC (Remote Function Call) is SAP's proprietary protocol for remotely executing function modules. Support for it arrived with version 1.3 of Cloud Connector. RFC with Secure Network Connection (SNC), the encrypted version of RFC, is also supported.

LDAP

Support for the LDAP (Lightweight Directory Access Protocol) protocol was added in Cloud Connector version 2.9. LDAP lets you communicate with on-premise directory services such as Microsoft's Active Directory. Cloud Connector also supports LDAPS, the encrypted version of LDAP.

TCP

TCP (Transmission Control Protocol) is the latest addition to Cloud Connector's range of supported protocols. Support for it arrived with version 2.10. With TCP, you can access any on-premise service listening for TCP connection requests on a given port number. An example of such a service is a mail server, communicating over the SMTP protocol. Cloud Connector also supports encrypted traffic over TCP. This option is referred to as TCP SSL. For more information on how to access backend resources over TCP, please see Section 4.9.

4.3 Adding a System Mapping

To make an on-premise resource available to the applications of an SAP Cloud Platform subaccount, you first need to add a mapping to the on-premise system that provides the resource and the protocol the resource is provided over.

Note

The term *mapping* refers to the fact that Cloud Connector lets you assign virtual connection information when you add a new on-premise system. So whenever you add an on-premise system, you are also adding a mapping from a virtual system to an internal system.

You add the mapping from the **Cloud To On-Premise** view located in the **Subaccount** menu, which is the UI element highlighted in Figure 25.

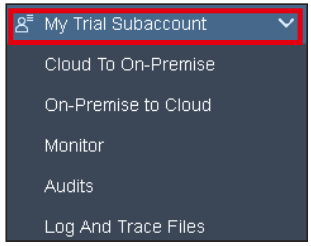


Figure 25 The Subaccount Menu in the Cloud Connector UI

Note how the menu's title indicates the currently selected subaccount. Any configuration you perform using the tools in this menu affects only this subaccount. All views in the **Subaccount** menu have a drop-down list of subaccounts at the top of the page. To switch to a different subaccount, just select the one you need from this list.

To add a new system mapping, go to the **ACCESS CONTROL** tab and click the **Add** icon + of the **Mapping Virtual To Internal System** table. This launches the **Add System Mapping** wizard.

> **Note**
>
> If you want to provide resources over multiple protocols from the same backend system, a mapping must be added for each protocol. For instance, an SAP system can provide access to OData services over HTTP and function modules over RFC. In that case, you need to add two mappings: one for the HTTP protocol and one for the RFC protocol.

The steps of the wizard are covered in the following sections.

Choose a Backend Type

In the first step of the wizard, shown in Figure 26, you choose the type of backend you want to access.

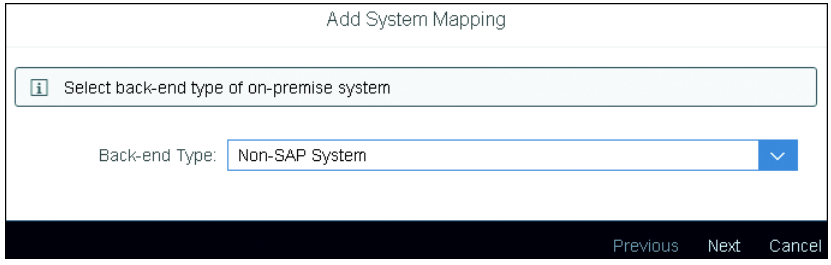


Figure 26 Choosing a Backend Type in the Add System Mapping Wizard

In the early days of Cloud Connector, the license limited how many non-SAP systems a customer could access. At that time, choosing **Non-SAP System** as the backend type would count against that limit. However, this is no

longer the case. Today, the chosen backend type only determines the protocols available in the next step of the wizard.

Choose the backend type that is the closest fit to the on-premise system you want to add, and click **Next**.

Choose a Protocol

In the next step of the wizard, shown in Figure 27, you choose the protocol you want to make resources available over.

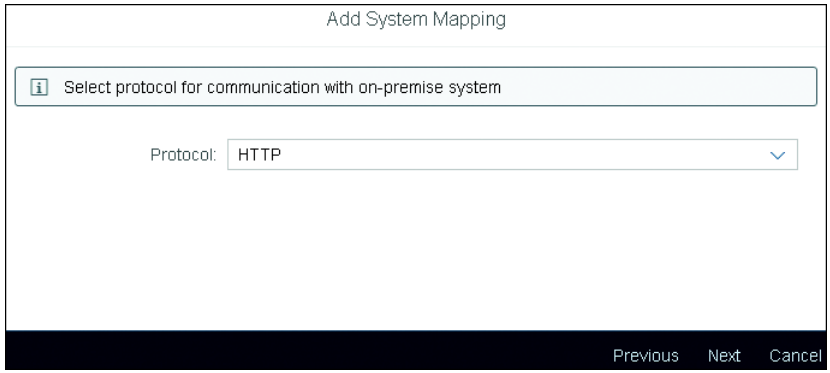


Figure 27 Choosing a Protocol in the Add System Mapping Wizard

The protocols available to you depend on which backend type you selected in the first step of the wizard. For instance, the RFC protocol is not available if you selected **Non-SAP System** as the backend type.

Choose the appropriate protocol from the drop-down list and click **Next**.

Note

Why does the wizard display a security warning, when you select the TCP protocol? This is because Cloud Connector does not know the application protocol, when an application communicates with a backend system over TCP. This means, for instance, that requests cannot be validated, and backend resources cannot be whitelisted. Security is entirely in the hands of the two communicating parties; Cloud Connector cannot help.

Choose a Load Balancing Option

The next step of the wizard, shown in Figure 28, is specific to RFC communications. It is displayed only if you selected either the RFC or the RFC with Secure Network Communications protocol in the previous step.

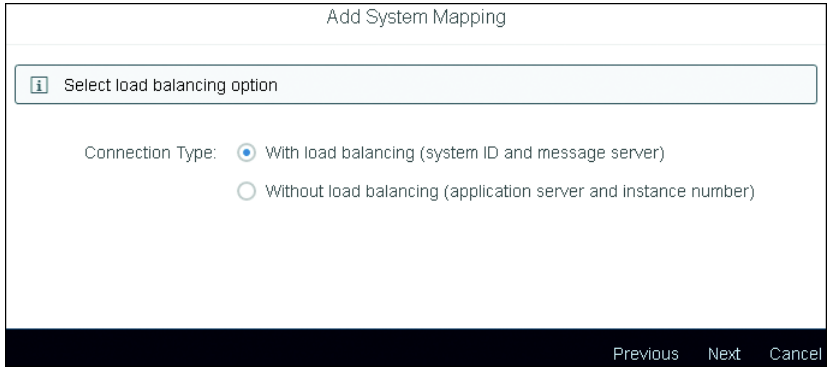


Figure 28 Choosing an RFC Load Balancing Option in the Add System Mapping Wizard

Indicate whether the connection to your SAP backend system is load balanced or not and click **Next**.

Enter the Connection Details

In the wizard's next step, you enter the connection details of the on-premise system. For all non-RFC protocols, the required information is the host name and port number, as shown in Figure 29.

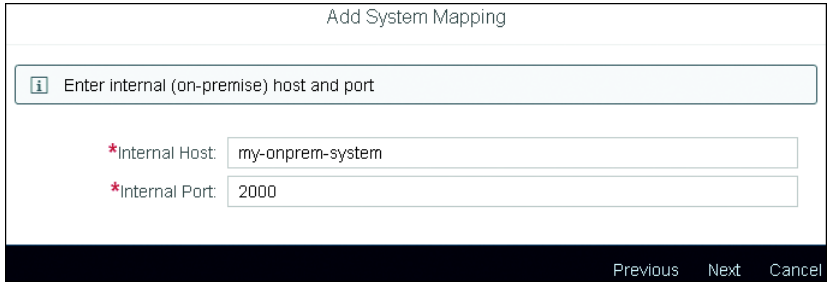


Figure 29 Entering a Non-RFC Connection's Details in the Add System Mapping Wizard

In the RFC case, the connection details depend on whether or not the connection is load balanced. For load balanced connections, shown in Figure 30, the required information is the message server and system ID or port number. If you need to connect through SAProuter, you have the option of entering an SAProuter string.

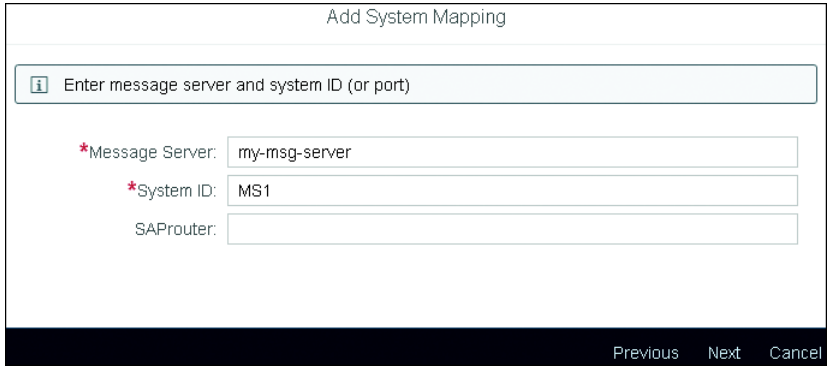


Figure 30 Entering the Connection Details for RFC with Load Balancing in the Add System Mapping Wizard

For RFC connections without load balancing, shown in Figure 31, the required information is the application server and instance or port number. If you need to connect through SAProuter, you have the option of entering an SAProuter string.

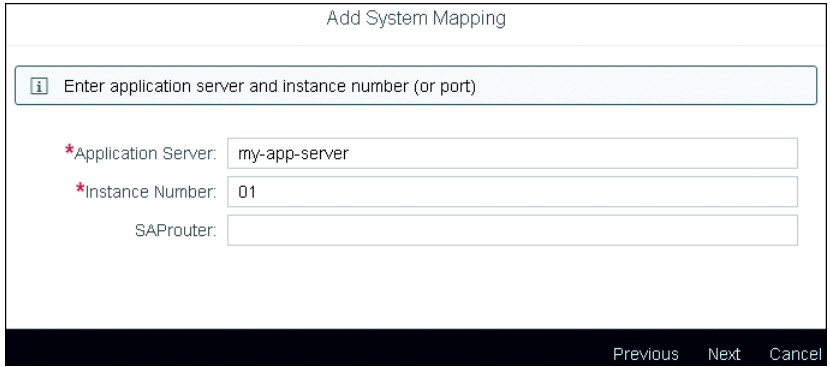


Figure 31 Entering the Connection Details for RFC without Load Balancing in the Add System Mapping Wizard

Note

Regardless of protocol, the on-premise system must be reachable over the network from the Cloud Connector host.

Enter the connection details required by the chosen protocol and click **Next**.

Enter the Virtual Connection Details

The wizard's next step lets you optionally assign virtual connection information to the on-premise system. If provided, SAP Cloud Platform applications and services will only be able to access the on-premise system through Cloud Connector using the virtual connection information.

As was the case in the previous step, the required virtual connection information depends on the chosen protocol. For all non-RFC protocols, the required information is a virtual host name and a virtual port number, as shown in Figure 32.

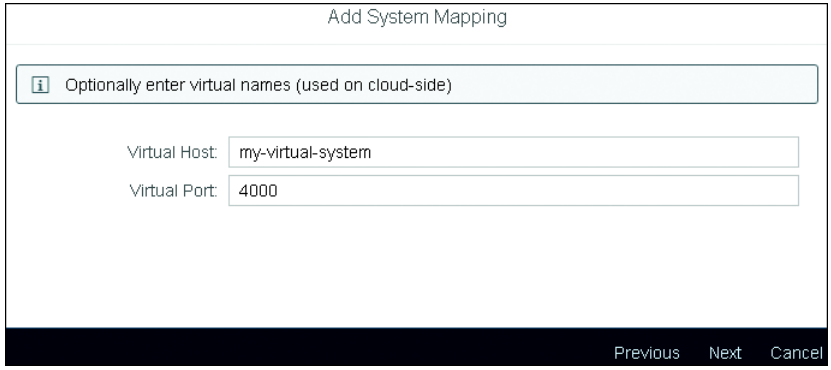


Figure 32 Entering a Non-RFC Connection's Virtual Connection Details in the Add System Mapping Wizard

For load balanced RFC connections, shown in Figure 33, the required virtual connection information is a virtual message server and a virtual system ID or port number.

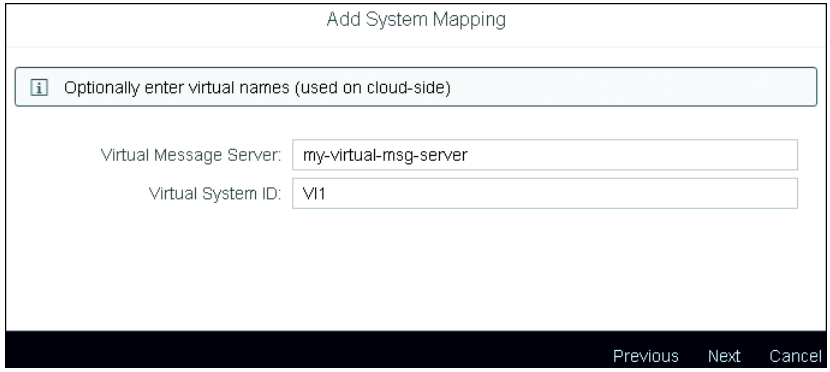


Figure 33 Entering the Virtual Connection Details for RFC with Load Balancing in the Add System Mapping Wizard

For RFC connections without load balancing, shown in Figure 34, the required virtual connection information is a virtual application server and a virtual instance or port number.

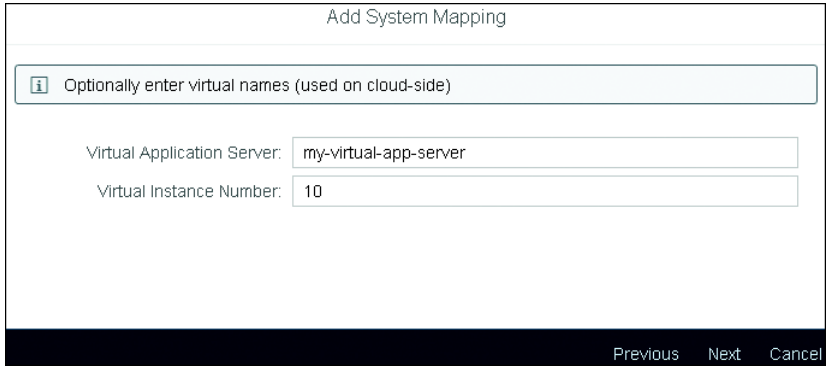


Figure 34 Entering the Virtual Connection Details for RFC without Load Balancing in the Add System Mapping Wizard

In all three cases, the values of the virtual connection input fields default to the real connection details, so if you don't need to provide virtual equivalents, you can leave them unchanged. Otherwise, enter the desired virtual connection details, as required by the chosen protocol, and click **Next**.

Choose a Principal Type

The wizard's next step, shown in Figure 35, is displayed only if the chosen protocol is either HTTP, HTTPS, or RFC with Secure Network Communications. It lets you choose how the user's identity will be propagated from SAP Cloud Platform to the backend system in a principal propagation scenario. For more information on principal propagation, please see Section 7.4.

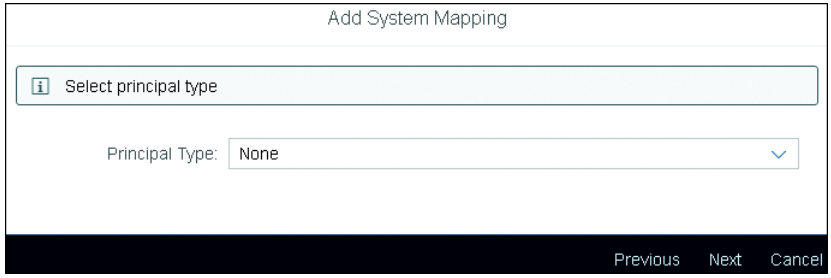


Figure 35 Choosing a Principal Type in the Add System Mapping Wizard

Select the appropriate principal type or **None**, if principal propagation is not in use, and click **Next**.

Enter the SNC Partner Name String

The next step of the wizard, shown in Figure 36, is specific to RFC with Secure Network Communications; it is displayed only if you chose that protocol in the wizard's second step.

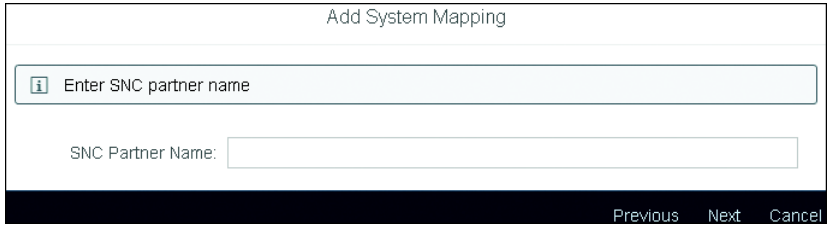


Figure 36 Entering the SNC Partner Name String in the Add System Mapping Wizard

Enter the SNC partner name string containing all the parameters required by your specific SNC library, and click **Next**.

Note

For more information on configuring SNC on the Cloud Connector side, please refer to Section 4.7.

Provide a Description

In the wizard's penultimate step, shown in Figure 37, you have the option of entering a description of the system mapping.

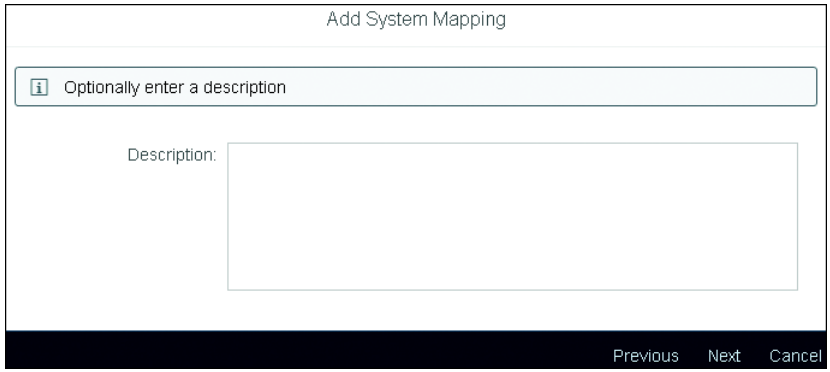


Figure 37 Entering a Description in the Add System Mapping Wizard

Since the description isn't mandatory, you can either provide one or leave the field empty before clicking **Next**.

Review the Summary and Conclude the Wizard

The last step of the wizard, shown in Figure 38, presents you with a summary of all the information you have entered.

If you want Cloud Connector to perform a check of whether it can reach the on-premise system over the network, select the **Check Internal Host** checkbox.

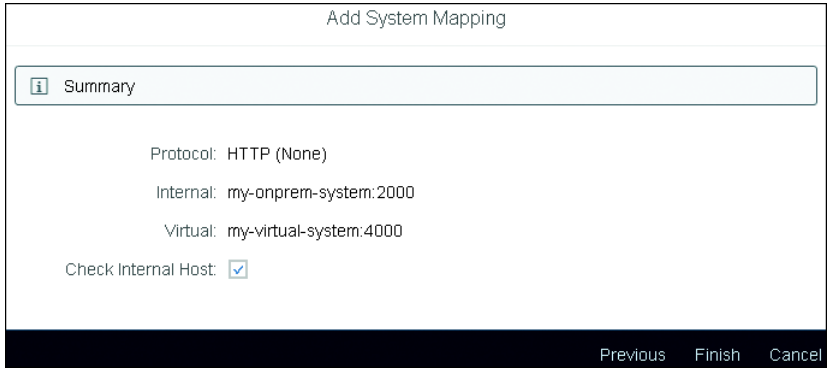


Figure 38 The Summary at the End of the Add System Mapping Wizard

If you are not happy with the summary, click **Previous** to navigate backward through the steps of the wizard and update the entered information as appropriate. Otherwise, click **Finish**. This concludes the wizard and adds the system mapping to the **Mapping Virtual To Internal System** table, as shown in Figure 39.

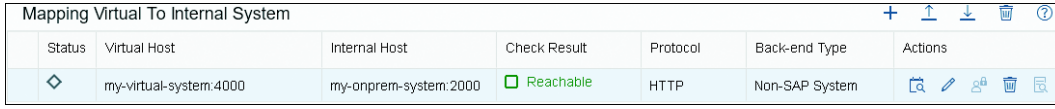


Figure 39 The New System Mapping Has Been Added to the Mapping Virtual To Internal System Table

In the example shown, Cloud Connector has checked the availability of the on-premise system and determined that it is reachable. You can perform the same check any time by clicking the availability check icon in the **Actions** column.

4.4 Adding an On-Premise Resource

Once an on-premise system has been added to a subaccount, you can start making that system's resources available to SAP Cloud Platform applications and services. With LDAP and TCP, you do not need to provide any

further information; only the host name and port number are required, and both were specified when you added the on-premise system. With HTTP and RFC resources, you need to provide some additional information.

To get started, go to the **ACCESS CONTROL** tab of the **Subaccount • Cloud To On-Premise** view. In the **Mapping Virtual To Internal System** table, select the system you want to add a resource from by clicking in the leftmost column of the system's row. This populates the **Resources Accessible On <host>** table with those of the system's resources that have already been added.

To add a new resource, click the **Add** icon + of the **Resources Accessible On <host>** table. This opens the **Add Resource** dialog. In the two following sections, we take a closer look at the **Add Resource** dialog for HTTP and RFC resources.

Adding an HTTP Resource

The **Add Resource** dialog for HTTP resources is shown in Figure 40.

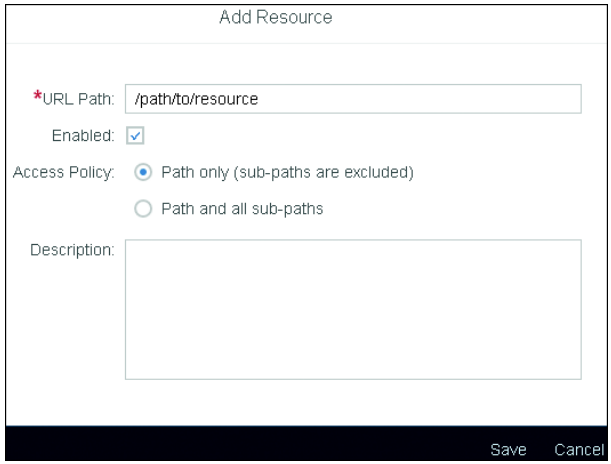


Figure 40 Adding an HTTP Resource

Enter the path to the HTTP resource into the **URL Path** field. The path is the part of the URL that comes after the host and port and before the parameters and named anchor. The URL path has been highlighted in Figure 41.

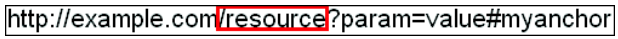


Figure 41 The Path Component of a URL

The **Enabled** checkbox indicates whether the resource should be enabled immediately after it has been added. If you don't want the resource to be enabled, clear the checkbox. Otherwise, leave it checked.

The **Access Policy** setting determines how Cloud Connector matches the path of an incoming request to the path you specified in the **URL Path** field. If **Path only** is selected, only an exact match will be accepted; all other paths will cause the request to fail and an entry to be written to the audit log (for more information on Cloud Connector's audit log, please see Section 7.7). If **Path and all sub-paths** is selected, Cloud Connector will accept all paths that start with the specified path.

Warning

The **Path and all sub-paths** option works essentially like a wildcard. Use it with caution so that you do not inadvertently circumvent the resource whitelisting security mechanism by allowing access to more resources than you intended.

Optionally enter a description of the HTTP resource in the **Description** field, and click **Save** to add the resource. This closes the dialog and adds the resource to the **Resources Accessible On <host>** table, as shown in Figure 42.

Figure 42 The New HTTP Resource Has Been Added to the Resources Table

Adding an RFC Resource

The **Add Resource** dialog for RFC resources is shown in Figure 43.

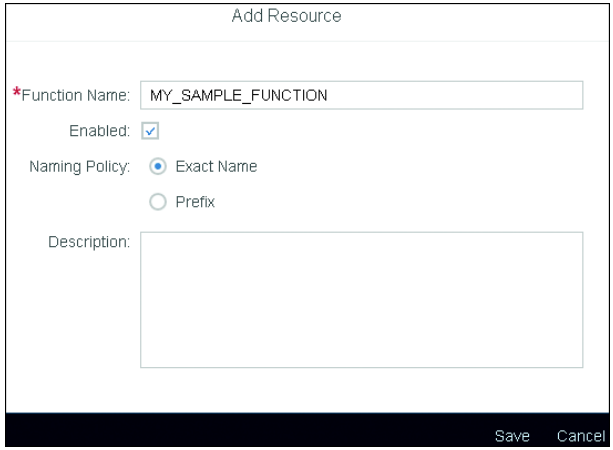


Figure 43 Adding an RFC Resource

The name of the function goes in the **Function Name** field.

The **Enabled** checkbox indicates whether the resource should be enabled immediately after it has been added. If you don't want the resource to be enabled, clear the checkbox. Otherwise, leave it checked.

The **Naming Policy** setting determines how the value of the **Function Name** field will be matched to an incoming request. If **Exact Name** is selected, the function module name of the request must match the provided name exactly, or the request will fail and an entry will be written to the audit log (for more information on Cloud Connector's audit log, please see Section 7.7). If **Prefix** is selected, Cloud Connector will accept all function module names that begin with the specified value.

Warning

The **Prefix** option works essentially like a wildcard. Use it with caution, so that you do not inadvertently circumvent the resource whitelisting security mechanism by allowing access to more resources than you intended.

Optionally, enter a description of the RFC resource in the **Description** field, and click **Save** to add the resource. This closes the dialog and adds the resource to the **Resources Accessible On <host>** table, as shown in Figure 44.

Figure 44 The New RFC Resource Has Been Added to the Resources Table

4.5 Mapping Cookie Domains

When a client makes an HTTP request to a web server, the response might contain a `Set-Cookie` response header, instructing the client to store a cookie, the contents of which is a name-value pair. The next time the client makes a request to the same web site, it will return the cookie in a `Cookie` request header.

When an SAP Cloud Platform application communicates with an HTTP backend system through Cloud Connector, the application knows only the backend system's virtual name. This does not impact the flow of cookies, though. If the backend system is called *api.example.com*, and its virtual name is *services.on-prem*, all cookies from the backend system will be stored by the application under the *services.on-prem* domain, and sent along whenever the application makes a request to that domain.

In general, cookies will work without any action needed on our part. There is one exception, though, which must be handled manually. The `Set-Cookie` header can contain a `Domain` attribute, indicating which domains this cookie should be returned to. Normally, a cookie originating from *api.example.com* will only be returned to that domain; it will not be sent along in a request to, say, *admin.example.com*. However, with the `Domain=example.com` attribute added to the `Set-Cookie` header, the cookie will be sent to *all* subdomains of *example.com*.

Passing the `Domain` attribute on to the SAP Cloud Platform application unchanged will not do. To the application, it will look like the server is trying

to store cookies in a domain that is not its own, which is not allowed. Therefore, the value of the `Domain` attribute must be translated in Cloud Connector.

If the backend system sets a cookie with the `Domain` attribute, and the value of the attribute matches the system's internal name (*api.example.com* in the above example), Cloud Connector can automatically translate the value to the system's virtual name (*services.on-prem* in the above example). However, if the value of the attribute is something else, we need to provide Cloud Connector with the correct mapping.

To add a cookie domain mapping, go to the **COOKIE DOMAINS** tab of the **Subaccount • Cloud To On-Premise** view, shown in Figure 45.

Figure 45 The List of Mapped Cookie Domains

To add a new domain mapping, click the **Add** icon + to open the **Add Domain Mapping** dialog, shown in Figure 46.

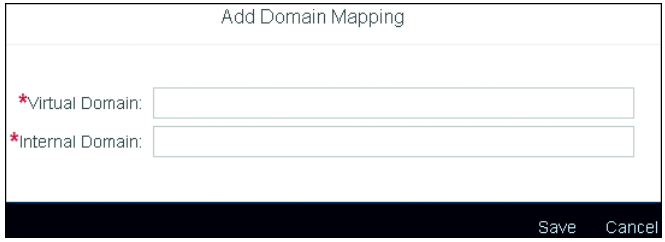


Figure 46 Adding a New Cookie Domain Mapping

Enter the replacement value into the **Virtual Domain** field, and the value to be replaced into the **Internal Domain** field. Then, click **Save** to store the mapping and close the dialog.

You can edit a domain mapping by clicking its **Edit** icon, and delete it by clicking its **Delete** icon. To delete all domain mappings, click the list's **Delete all mappings** icon.

4.6 Additional RFC Protocol Security

In addition to whitelisting the function modules that SAP Cloud Platform applications and services will be able to call, Cloud Connector supports two more security measures for the RFC protocol: whitelisting clients and blacklisting users.

To manage these two lists, click the **Maintain authority lists (RFC only)** icon in the **Actions** column of the appropriate system's row in the **Mapping Virtual To Internal System** table. This opens the **Edit Authority for Client and User** dialog, shown in Figure 47.

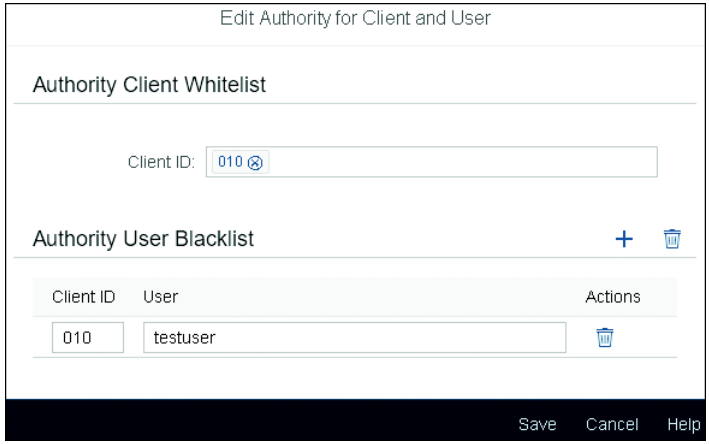


Figure 47 Managing the Client Whitelist and the User Blacklist

Enter the client numbers that you wish to whitelist into the **Client ID** field in the **Authority Client Whitelist** section, pressing Enter after each client number. As soon as at least one client has been whitelisted, any request to a client *not* on this list will fail.

To blacklist a user, click the **Add a user authority** icon + in the **Authority User Blacklist** section. Enter the client number into the **Client ID** field and the user name into the **User** field. Any request that authenticates with a client number and user name combination found on this list will fail.

When you are done managing the lists, click **Save** to store them.

4.7 Configuring Cloud Connector for SNC

Before you can use RFC with Secure Network Communications protection, you need to configure Cloud Connector for SNC. To do so, go to the **ON PREMISE** tab in the **Connector • Configuration** view and scroll to the **SNC** section, shown in Figure 48.

Figure 48 The SNC Section of the On-Premise Configuration Tab

To edit the SNC configuration, click the **Edit** icon to open the **Edit SNC** dialog, shown in Figure 49.

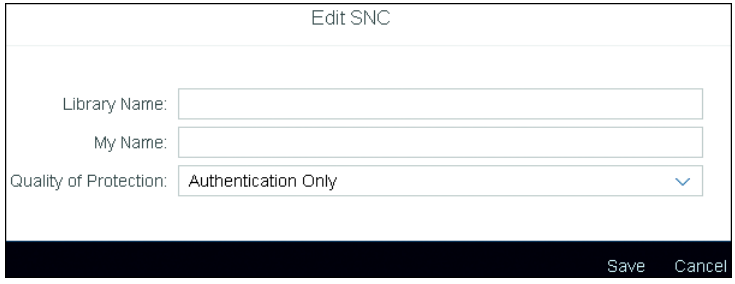


Figure 49 The Edit SNC Dialog

The **Library Name** field contains the path to the SNC library on the Cloud Connector host.

The **My Name** field contains Cloud Connector's partner name string, containing all the parameters that your specific SNC library requires.

The **Quality of Protection** drop-down list contains the available protection levels.

To configure Cloud Connector for SNC, provide the path to the SNC library, fill out Cloud Connector's partner name string and choose a protection level. Then, click **Save** to store the SNC configuration and close the dialog.

4.8 Hands-on: Calling an On-Premise REST Service from the Cloud

In this hands-on exercise, you will set up a REST service running on your local machine and then proceed to call that service from an SAP Cloud Platform HTML5 application through Cloud Connector. There are four steps, and they mirror the steps you will take when you configure a similar scenario in a production environment.

The REST service returns the first 100 natural numbers along with the primality of each number. The HTML5 application retrieves this information by calling the service and then formats it for display in a table.

To complete the exercise, you will need the E-Bite's supplemental materials. You can download them from the E-Bite's page in your e-book library on www.sap-press.com, under **Additional Data**. You will find the files for this exercise in the directory named *section-4*.

Install Apache Tomcat and Deploy the REST Service

The REST service is a Java 8 servlet, so to run it, you need Java 8 and a servlet container installed on your machine. There are several servlet containers available, but the rest of the exercise assumes that you are running Apache Tomcat.

Installing Apache Tomcat is straightforward. Just go to the Apache Tomcat website (tomcat.apache.org), download the latest version, and follow the directions given in the *RUNNING.txt* file, which you will find in the root of the downloaded archive.

The *RUNNING.txt* file also describes how to launch Apache Tomcat once the initial configuration has been completed. When the servlet container is running, go to http://localhost:8080/ in your browser to verify the installation. If everything is running correctly, you will see a page similar to the one shown in Figure 50.

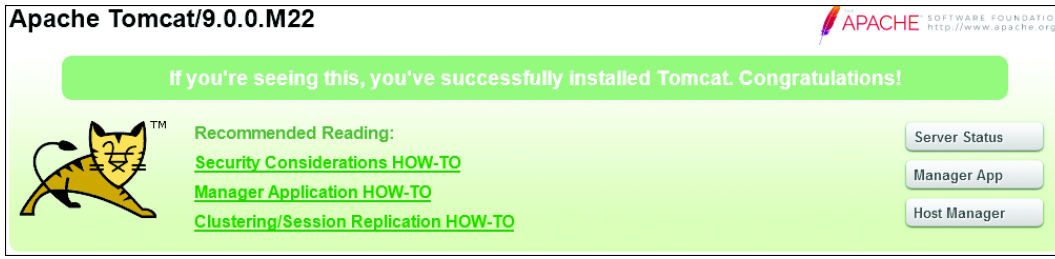


Figure 50 Apache Tomcat Has Been Successfully Installed

When Apache Tomcat is up and running, you are ready to deploy the web application containing the REST service. From the E-Bite's supplemental materials, copy the *sccbook.war* file from the *section-4* directory into Apache Tomcat's *webapps* directory. Apache Tomcat will extract the archive's contents and deploy the application automatically.

To verify the deployment, go to localhost:8080/sccbook/primes in your browser. If you see JSON data similar to Figure 51, the web application has been deployed correctly.

Note

Some browsers, such as Google Chrome, display the output as raw JSON, while others, such as Mozilla Firefox, render it.

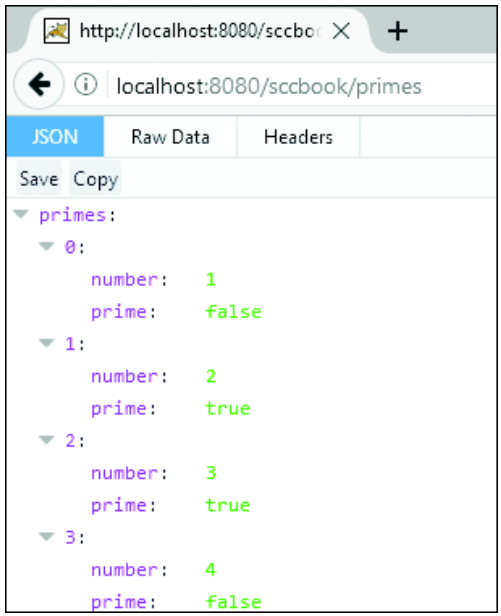


Figure 51 Output of the Primes REST Service as Seen in the Firefox Browser

Configure Cloud Connector

With the REST service in place, we can add it to Cloud Connector as an HTTP resource. First, create a system mapping to port 8080 on your local machine. Launch the **Add System Mapping** wizard, as described in Section 4.3, and enter the values listed in Table 2.

Setting	Value
Backend Type	Non-SAP System
Protocol	HTTP
Internal Host	localhost
Internal Port	8080
Virtual Host	primes-server
Virtual Port	4000
Principal Type	None

Table 2 Adding a System Mapping to Port 8080 of Your Local Machine

Port 8080 is Apache Tomcat's default port. If your installation uses a different port, enter that port number in place of 8080.

Next, add the REST service as an HTTP resource, as described in Section 4.4, using the values listed in Table 3.

Setting	Value
URL Path	/sccbook/primes
Enabled	Checked
Access Policy	Path only

Table 3 Adding the REST Service as an HTTP Resource

When you are done, the system mapping and resource should be listed on the **ACCESS CONTROL** tab, similar to what you see in Figure 52.

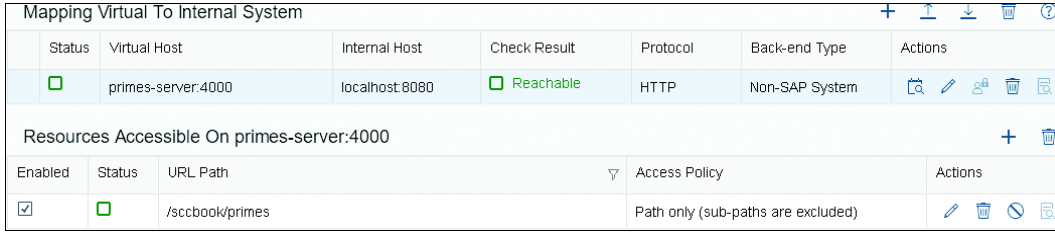


Figure 52 The System Mapping and Resource Have Been Created Successfully

Create an SAP Cloud Platform Destination

To call the on-premise service from an HTML5 application, we need to create an SAP Cloud Platform destination pointing to it. To do so, go to the **Connectivity • Destinations** view in the SAP Cloud Platform cockpit and click **Import Destination**. In the file selector dialog, choose the *primes_dest* file in the *section-4* directory and click the **Save** button to store the destination.

Figure 53 shows the new destination's row in the table of destinations, once it has been imported.

		Authentication	NoAuthentication
HTTP	primes_dest	ProxyType	OnPremise
		URL	http://primes-server:4000/sccbook/primes

Figure 53 The Destination Has Been Successfully Imported

Note that the host name and port number are the virtual values configured in Cloud Connector in the previous step and that the proxy type is **OnPremise**, indicating that requests to this destination should be routed through Cloud Connector.

Import and Run the HTML5 Application

In the last step of the exercise, we import the HTML5 application into the Web IDE and run it.

> **Note**
>
> To keep the hands-on exercise short, we will skip deploying the application to SAP Cloud Platform, and instead run it from inside the Web IDE.

If this is your first time launching the Web IDE, go to the **Services** view in the SAP Cloud Platform cockpit, locate the Web IDE service in the **DevOps** section, and click it. On the service overview page, click the **Enable** button to enable the service, and then click the **Go to Service** link to launch the Web IDE. For easier access in the future, consider adding the Web IDE to your bookmarks.

Once you are in the Web IDE, choose **File • Import • From File System**. In the **Import** dialog, browse to the *PrimesApp.zip* file in the *section-4* directory. Leave the other settings at their default values and click **OK**. The HTML5 application is now imported into the Web IDE. Figure 54 shows the structure of the application.

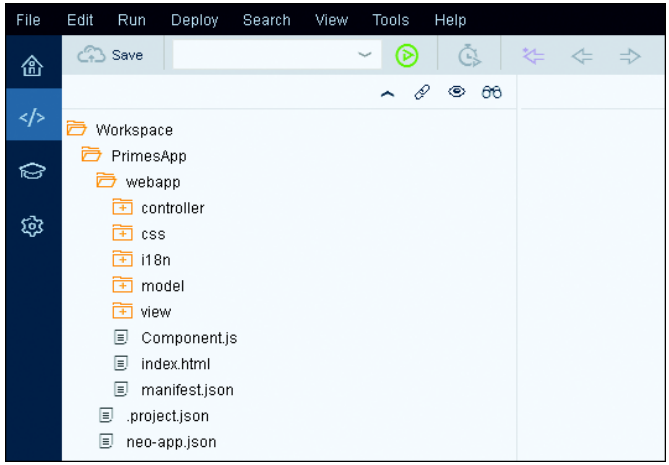


Figure 54 The HTML5 Application Has Been Imported into the Web IDE

Having imported the application, we are now ready to run it. To do so, select any file or folder within the application project and click the **Run** icon in the Web IDE toolbar. Figure 55 shows the application running in the browser.

Primes

Number	Prime?
1	No
2	Yes
3	Yes
4	No
5	Yes
6	No
7	Yes
8	No
9	Yes
10	No

Figure 55 The Running HTML5 Application

To verify that the displayed data really does originate from the on-premise REST service, go to the **MOST RECENT REQUESTS** tab of the **Subaccount • Monitor** view in the Cloud Connector UI. Here you will see an entry similar to the one shown in Figure 56.

Date	Virtual Host	Protocol	Duration (ms)	Actions
Jul 16, 2017 1:30:07 PM	primes-server:4000	HTTP	5	

Figure 56 A Logged Request to the On-Premise REST Service

To learn more about monitoring Cloud Connector traffic, please see Section 6.1.

4.9 Accessing a TCP Resource from SAP Cloud Platform

Accessing a backend resource over TCP differs somewhat from how you access resources over the other protocols that Cloud Connector supports. To connect to an on-premise service listening for TCP requests, you must go through a SOCKS5 proxy server provided by SAP Cloud Platform.

From your SAP Cloud Platform application, the proxy can be accessed using host name *localhost* and port number 20004. The application must run server-side, however, and in SAP Cloud Platform's Neo environment.

The SOCKS5 server needs some information about the Cloud Connector instance to correctly route requests to it. Specifically, it needs the subaccount that the instance is connected to and the location ID (the default location ID is represented by the empty string). These two values must be Base64 encoded and placed in the user name part of the basic authentication credentials in the format `1.<subaccount>.<locationID>`.

The sample Java code in Listing 1 demonstrates how to set up the authentication and obtain a `java.net.Proxy` object representing the SOCKS5 proxy.

```
import java.util.Base64;
import java.net.Authenticator;
import java.net.PasswordAuthentication;
import java.net.SocketAddress;
import java.net.InetSocketAddress;
import java.net.Proxy;

// Set basic proxy authentication
Base64.Encoder b64 = Base64.getEncoder();
String subaccountBase64 = b64.encodeToString("my-
subaccount".getBytes());
String locationIdBase64 = b64.encodeToString("my-location-
ID".getBytes());
Authenticator.setDefault(new Authenticator() {
    @Override
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication("1." + subaccountBase64
          + "." + locationIdBase64, new char[]{});
    }
});
// Create the SOCKS5 proxy object
SocketAddress proxyAddr =
 new InetSocketAddress("localhost", 20004);
Proxy proxy = new Proxy(Proxy.Type.SOCKS, proxyAddr);
```

Listing 1 Getting a SOCKS5 Proxy Object in Java

If you use this code in your own application, remember to update the dummy subaccount and location ID values.

How you use the `java.net.Proxy` object once you have obtained it depends on the on-premise service you are accessing. The sample Java code in Listing 2 shows how to open a socket to an on-premise server through the SOCKS5 proxy.

```
import java.net.Socket;
import java.net.SocketAddress;
import java.net.InetSocketAddress;
```

```
Socket socket = new Socket(proxy); // proxy is a java.net.Proxy
                                   // object
SocketAddress endpoint = new InetSocketAddress("on-premise-
server", 8000);
socket.connect(endpoint);
```

Listing 2 Opening a Socket to an On-Premise Server through the SOCKS5 Proxy

In the sample code, a socket is opened to port 8000 of the host named *on-premise-server*. The connection request will be routed through Cloud Connector by the SOCKS5 proxy.

5 Configuring On-Premise to Cloud Access

In the previous section, you learned how to configure Cloud Connector to provide access to on-premise resources from the cloud. In this section, we cover the other Cloud Connector scenario: providing access to cloud resources from on-premise.

First, we will take a closer look at the on-premise to cloud scenario. Then, you will learn how to access SAP HANA databases in your SAP Cloud Platform subaccount from on-premise tools via database service channels. Next, we will cover how to connect to a virtual machine running on SAP Cloud Platform using SSH through a virtual machine service channel. Finally, you will learn how to call SAP S/4HANA Cloud function modules from on-premise systems by way of an S/4HANA Cloud service channel.

5.1 On-Premise to Cloud Overview

The on-premise to cloud scenario is all about letting your on-premise systems and applications securely access cloud resources that would otherwise be inaccessible. The scenario's central concept is that of the *service channel*. A service channel makes a cloud resource available in your on-premise

environment through Cloud Connector. At the time of writing, Cloud Connector supports three service channels: a service channel for SAP HANA databases, a service channel for virtual machines, and a service channel for SAP S/4HANA Cloud function modules. In the following sections, you will learn more about each available service channel.

Service channels are configured in the **Subaccount • On-Premise to Cloud** view. The **Service Channels** section, shown in Figure 57, contains a list of every service channel that is currently configured.

Figure 57 The List of Currently Configured Service Channels

The **Actions** column contains icons for editing, deleting, enabling, and disabling that row's service channel.

The list of service channels has an **Add** icon for adding a new service channel and a **Delete** icon for removing every configured service channel.

5.2 Database Service Channels

In order to connect an on-premise database tool to an SAP HANA database running in your SAP Cloud Platform subaccount, a secure tunnel must be established.

If you are using the SAP HANA Eclipse tool, such a tunnel is automatically opened and closed for you as needed. However, to use other database tools, such as SQL clients, ETL tools, or replication tools, you need to set a tunnel up yourself. You can do so in Cloud Connector by creating a database service channel.

Note

An alternative, non-Cloud Connector way to create a database tunnel is to do so from the command line using the `open-db-tunnel` command of the SAP Cloud Platform console client.

When you have created a database service channel, an on-premise database tool can connect to it using ODBC or JDBC, as if it was the actual SAP HANA database in the cloud.

In the following sections, we cover how to create a Cloud Connector database service channel and how to connect your database tool of choice to SAP HANA in the cloud via a database service channel.

Creating a Database Service Channel

To create a database service channel, click the **Add** icon + in the **Subaccount • On-Premise to Cloud** view. This launches the **Add Service Channel** wizard. In the wizard's first step, shown in Figure 58, choose **HANA Database** from the **Type** drop-down list and click **Next** to proceed to the next step.

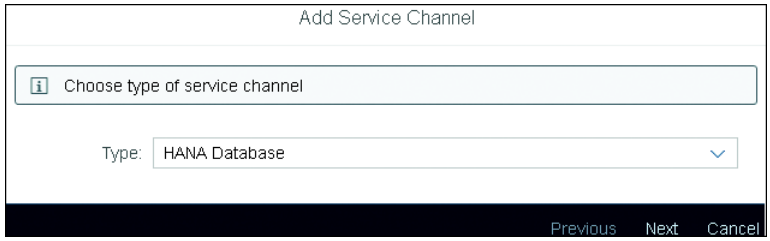


Figure 58 Choosing the Service Channel Type in the Add Service Channel Wizard

In the second step of the wizard, shown in Figure 59, you provide the instance name of the SAP HANA database and a local instance number.

Select the instance name from the **HANA Instance Name** drop-down list, or enter the name into the field manually if the instance name is not in the list.

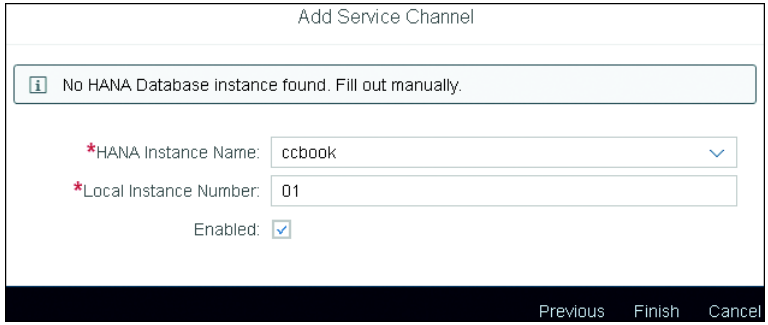


Figure 59 Providing an SAP HANA Instance Name and a Local Instance Number in the Add Service Channel Wizard

The **Local Instance Number** field contains a two-digit number between 01 and 99. The local instance number determines the port number that a database tool must connect to. The port number can be determined as follows: `3<local instance number>15`.

> **Note**
>
> The local instance number is independent of the instance number of the SAP HANA database running in your SAP Cloud Platform subaccount. The instance numbers can be the same, but they do not have to be.

To enable the database service channel immediately after its creation, leave the **Enabled** checkbox selected. To manually enable it later, clear the checkbox. Then click **Finish** to conclude the wizard and create the database service channel.

Connecting Your Database Tool

With the database service channel in place, you are ready to connect your database tool to the SAP HANA database running in your SAP Cloud Platform subaccount.

Depending on your specific tool, you will need either an ODBC driver or a JDBC driver. Both are delivered as part of the SAP HANA client installation.

If your database tool connects via JDBC, use the following JDBC URL:

```
jdbc:sap://<cloud connector host>:<database service channel port>
[/?<options>]
```

If your database tool connects via ODBC, use the following ODBC connection string:

```
DRIVER=HDBODBC;UID=<hana user>;PWD=<hana user password>;SERVERNODE=
<cloud connector host>:<database service channel port>
```

Note

To use the 32-bit ODBC driver instead of the 64-bit driver, replace `HDBODBC` with `HDBODBC32`.

5.3 Virtual Machine Service Channels

A virtual machine running in your SAP Cloud Platform subaccount cannot be accessed from the outside world. If you want to connect to it using an on-premise SSH client, you need to establish a secure tunnel to it first. You can do so in Cloud Connector by creating a virtual machine service channel.

Note

An alternative, non-Cloud Connector way to create an SSH tunnel is to do so from the command line using the `open-ssh-tunnel` command of the SAP Cloud Platform console client.

When a virtual machine service channel has been created, an on-premise SSH client can connect to it as if it was the actual virtual machine in the cloud.

In the following section, you will learn how to create a virtual machine service channel in Cloud Connector.

Creating a Virtual Machine Service Channel

To create a virtual machine service channel, click the **Add** icon + in the **Subaccount • On-Premise to Cloud** view to launch the **Add Service Channel** wizard. In the first step of the wizard, shown above in Figure 58, choose **Virtual Machine** from the **Type** drop-down list and click **Next** to proceed.

In the wizard's second step, shown in Figure 60, you choose which virtual machine the service channel should connect to and its local port number.

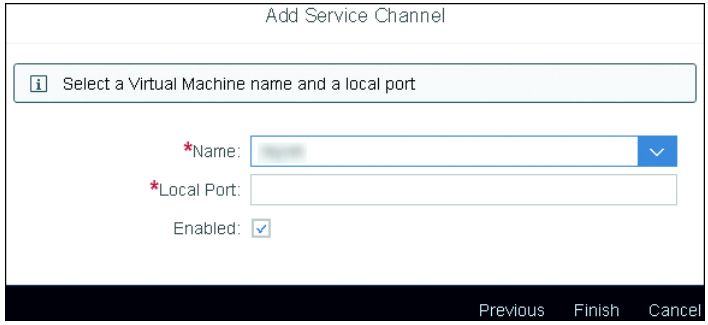


Figure 60 Choosing a Virtual Machine and Local Port in the Add Service Channel Wizard

Select the name of the virtual machine from the **Name** drop-down list, which is populated with the names of all the virtual machines created in your SAP Cloud Platform subaccount.

The **Local Port** field contains the port number that your SSH client will connect to. You can enter any port number, provided it is not already in use on the Cloud Connector host.

To enable the virtual machine service channel immediately after it has been created, leave the **Enabled** checkbox selected. To manually enable it later, clear the checkbox. Then click **Finish** to conclude the wizard and create the virtual machine service channel.

Once the virtual machine service channel is up and running, you can connect to the virtual machine in the cloud using an SSH client of your choice. To do so, you must provide the SSH client with the Cloud Connector host

name, the port number of the virtual machine service channel, and the key file that was generated when the virtual machine was originally created.

5.4 S/4HANA Cloud Service Channels

By default, you cannot call function modules on an SAP S/4HANA Cloud tenant via RFC from an on-premise system. However, with a Cloud Connector S/4HANA Cloud service channel in place, you can.

When an S/4HANA Cloud service channel has been created, an on-premise system can communicate with it using RFC as if it was the actual SAP S/4HANA Cloud tenant.

In the following section, we cover how to create an S/4HANA Cloud service channel.

Creating an S/4HANA Cloud Service Channel

To create an S/4HANA Cloud service channel, click the **Add** icon + in the **Subaccount • On-Premise to Cloud** view. This launches the **Add Service Channel** wizard. In the first step of the wizard, shown above in Figure 58, choose **S/4HANA Cloud** from the **Type** drop-down list and click **Next**.

In step two of the wizard, shown in Figure 61, you provide the SAP S/4HANA Cloud tenant host and a local instance number.

The **S/4HANA Cloud Tenant Host** field contains the host name of your SAP S/4HANA Cloud tenant.

In the **Local Instance Number** field, enter the instance number that an on-premise system will later use to call function modules via the service channel.

If you want the S/4HANA Cloud service channel to be enabled immediately after it has been created, leave the **Enabled** checkbox selected. To manually enable it later, clear the checkbox. Then click **Finish** to complete the wizard and create the S/4HANA Cloud service channel.

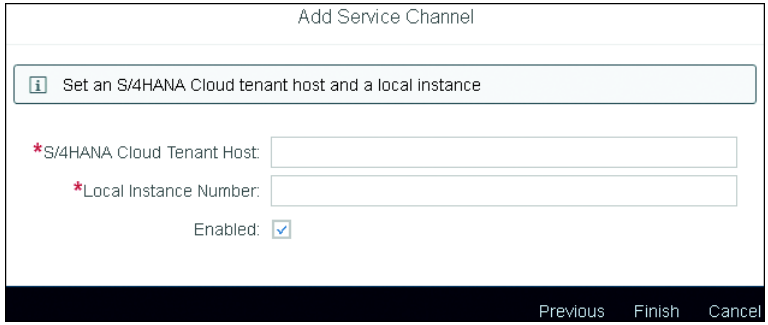


Figure 61 Providing an SAP S/4HANA Cloud Tenant and a Local Instance Number in the Add Service Channel Wizard

When the S/4HANA Cloud service channel is up and running, an on-premise system can use it to call function modules on your SAP S/4HANA Cloud tenant. The required configuration information is the Cloud Connector host name and the local instance number.

6 Cloud Connector Operations

In the previous sections, you learned how to install Cloud Connector, how to connect a Cloud Connector instance to an SAP Cloud Platform subaccount, and how to configure Cloud Connector to provide access to both on-premise and cloud resources. In this section, we will cover Cloud Connector operations.

You will learn how to monitor the traffic going through Cloud Connector and how logging and tracing works. You will also learn how to back up your configuration and how to restore such a backup. We will cover the APIs offered by Cloud Connector, which you can use to, for instance, monitor Cloud Connector from your existing monitoring solution.

Next, you will learn how Cloud Connector's alerting works, how to configure what you want to be alerted about, and how to receive alerts via email.

Cloud Connector's high availability setup will also be covered, as will how the high availability setup can minimize downtime during upgrades.

Finally, we will cover how to change the port number of the Cloud Connector UI.

6.1 Monitoring Cloud Connector Traffic

The **Subaccount • Monitor** view offers four tools for monitoring the traffic that passes through Cloud Connector and Cloud Connector's connections to backend systems. In the following sections, you will learn how to use each tool.

> **Note**
>
> The traffic tools show only HTTP and RFC requests.

Performance

The **PERFORMANCE** tab, shown in Figure 62, contains a bar chart of all recorded requests grouped into buckets based on the processing time of each request.

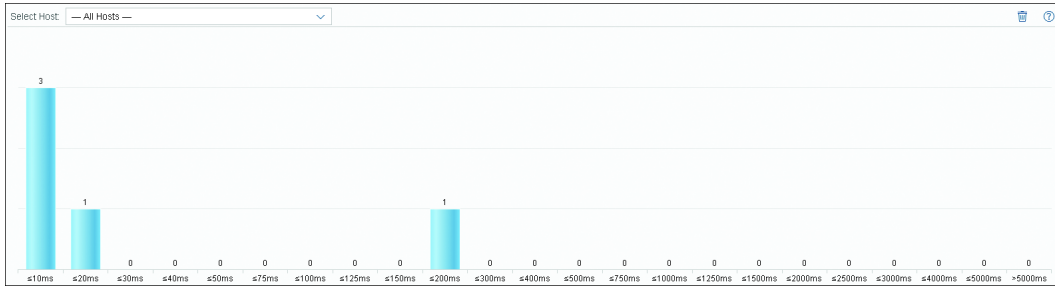


Figure 62 The Performance Monitoring Tab

Each bucket represents an interval of processing times. In the example shown in Figure 62, three requests took less than 10 milliseconds to process,

one request took between 10 and 20 milliseconds, and one request took between 150 and 200 milliseconds.

By default, the bar chart is based on requests to all hosts. To limit it to requests to a specific destination host, choose that host from the **Select Host** drop-down list.

The performance data that the bar chart is based on can be reset at any time. To do so, click the **Delete all performance data** icon. Be aware, however, that since the **MOST RECENT REQUESTS** and **TOP TIME CONSUMERS** monitoring tabs are based on the same data, deleting that data will also clear those two tabs.

Most Recent Requests

The **MOST RECENT REQUESTS** tab, shown in Figure 63, displays the most recent requests that the Cloud Connector instance has processed. Each request is listed with its date and time, the destination host, the request's protocol, and the time it took to process it. The list is sorted by the requests' date and time, with the most recent requests listed first.

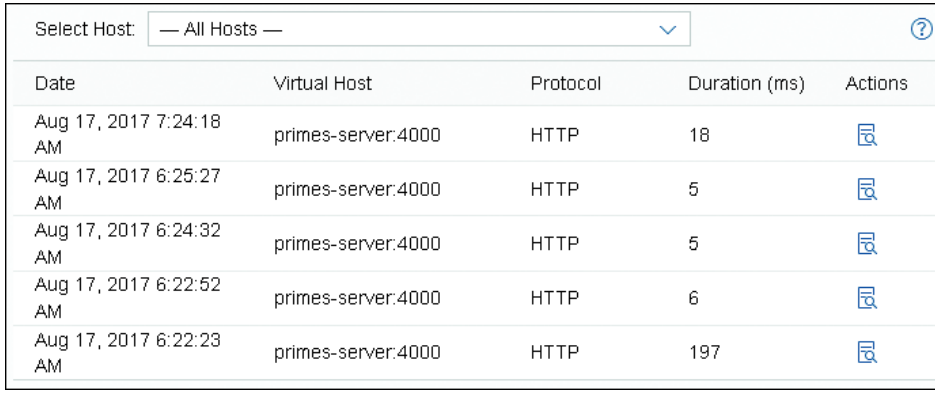


Figure 63 The Most Recent Requests Monitoring Tab

By default, the list contains requests to all hosts. To limit it to requests to a specific destination host, choose that host from the **Select Host** drop-down list.

Note

The list of most recent requests is capped at 50 entries. When the list is full, additional requests will push older requests off the list.

To see the details of a particular request, click the **Show details** icon in the **Actions** column of that request's row. This opens the **Request Details** dialog, shown in Figure 64.

Figure 64 The Request Details Dialog Displays the Details of a Single Request

The **Basic Data** section displays basic information about the request, such as the protocol, the requested resource, and bytes sent and received. The **Duration Breakdown** section visually breaks down how the request's processing time was spent.

Top Time Consumers

The **TOP TIME CONSUMERS** tab, shown in Figure 65, displays the same information as the **MOST RECENT REQUESTS** tab. However, the list is sorted by the requests' processing time, with the longest-running requests listed first. This lets you easily spot any requests that took a disproportionately long time to process.

By default, the list contains requests to all hosts. To limit it to requests to a specific destination host, choose that host from the **Select Host** drop-down list.

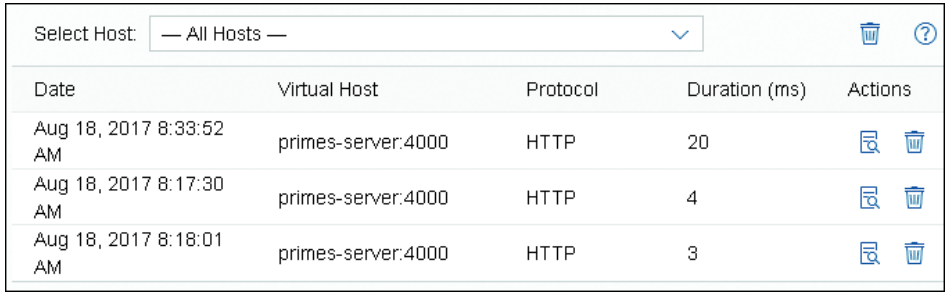


Figure 65 The Top Time Consumers Monitoring Tab

The **Show Details** icon in the **Actions** column opens the **Request Details** dialog described in the previous section. To remove a request from the list, click the **Delete this top time consumer** icon in the **Actions** column of that request's row.

To empty the list completely, click the list's **Delete all data related to top time consumers** icon. This does not impact the **PERFORMANCE** and **MOST RECENT REQUESTS** tabs.

Backend Connections

The **BACK-END CONNECTIONS** tab, shown in Figure 66, displays the current number of active and idle connections, grouped by virtual host and protocol.

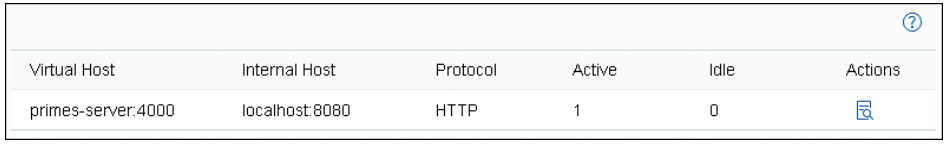


Figure 66 The Back-end Connections Monitoring Tab

To view the connection details of a virtual host and protocol, click that row's **Show details** icon. This opens the **Back-end Connection Details** dialog, shown in Figure 67.

In the **Active Connections** section, each active connection is listed with the date and time the connection became active, the user name (if applicable), and the backend resource. The **Idle Connections** section visually displays the number of idle connections over time.

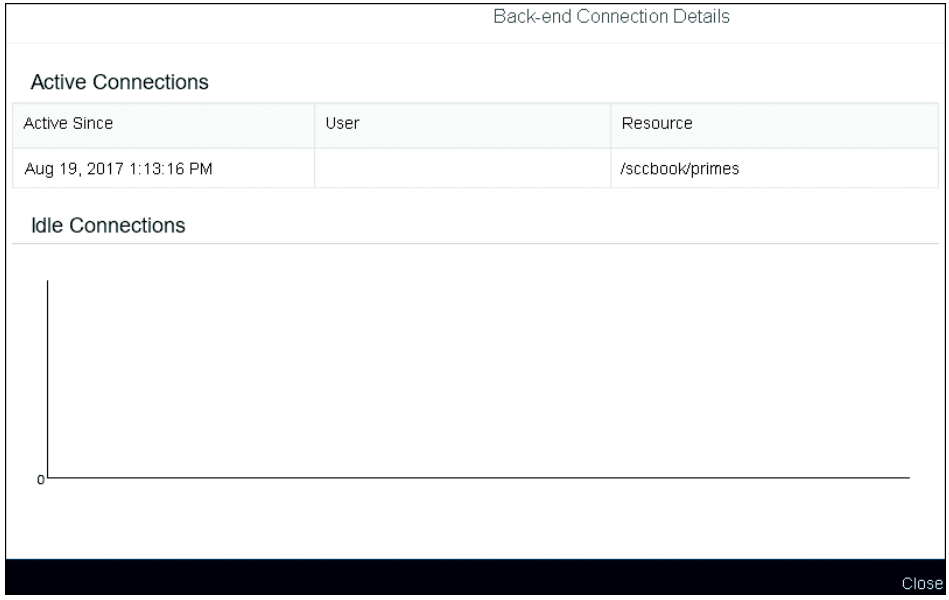


Figure 67 The Active and Idle Connections of a Virtual Host and Protocol

6.2 Logs and Traces

Cloud Connector maintains several log and trace files containing detailed information about its inner workings. The files are mainly intended for SAP Support's use, but they can also help you in debugging Cloud Connector issues.

The log and trace files are accessible from the **Subaccount • Log And Trace Files** view, shown in Figure 68.

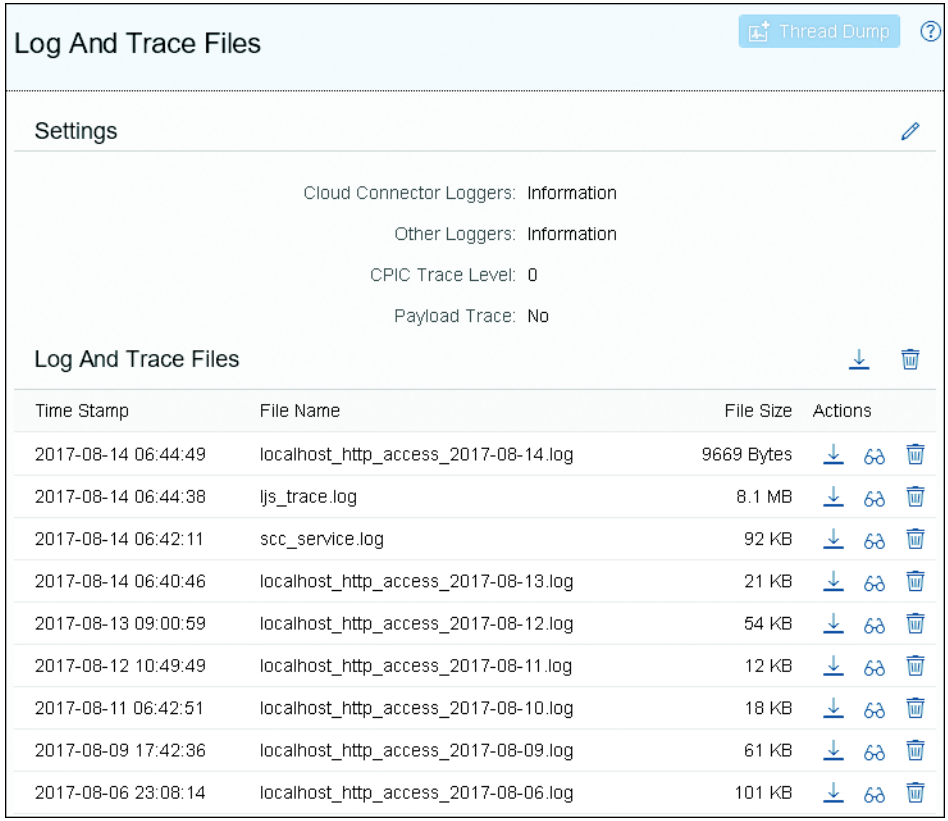


Figure 68 The Log And Trace Files View

Note

The **Log And Trace Files** view is located in the subaccount menu, because it contains both files, that are specific to the current subaccount, and files, that are independent of subaccounts. An example of the former is the payload trace files. An example of the latter is the HTTP access logs.

Each file is listed with the date and time of its last update, the file's name, and its size. The list is sorted by the files' time stamps, with the most recently updated files listed first. The **Actions** column contains three icons, which you can click to download, view, or delete that row's file.

The list also has icons for downloading every listed file in a compressed archive and deleting them all at once.

Note

Cloud Connector will not you let delete a file that is currently in use.

When you click the **Display** icon to view a log or trace file, Cloud Connector will open the file in the log viewer, displaying one page of text at a time. The log viewer's navigation bar, shown in Figure 69, lets you switch to the first, last, previous, and next page. You can also click and drag the gray handle along the horizontal slider or click anywhere inside the slider to jump to a specific region of the file.

Figure 69 The Log Viewer Navigation Bar

If you are experiencing an issue with Cloud Connector, your first step in diagnosing the problem should be to check the tail end of the *ljs_trace.log* file, which is Cloud Connector's default trace. In some cases, you might need to increase the level of detail written to the trace file by increasing the *log level*. To do so, click the **Edit** icon in the **Settings** section. This opens the **Edit Connector Info** dialog, shown in Figure 70.

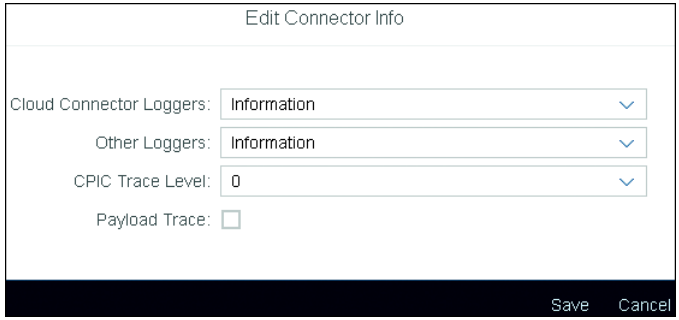


Figure 70 Configuring What Will Be Written to Cloud Connector's Log and Trace Files

The log level selected in the **Cloud Connector Loggers** drop-down list affects the logging related to Cloud Connector's functionality.

The log level selected in the **Other Loggers** drop-down list affects the logging performed by other Java loggers. It is only necessary to change this log level if SAP Support requests that you do so.

The log level selected in the **CPIC Trace Level** drop-down list affects the logging related to RFC communications between Cloud Connector and ABAP-based backend systems. This setting applies to RFC communications in all subaccounts, not just the current subaccount

Selecting the **Payload Trace** checkbox will cause HTTP and RFC traffic from the current subaccount to be stored in subaccount-specific trace files.

Warning

Logging network traffic, which you do by selecting the **Payload Trace** checkbox or setting the CPIC log level to 3, can potentially cause sensitive business and security information to be stored on disk. SAP recommends only doing so when requested to by SAP Support. For more information on securing traffic traces, please refer to the Cloud Connector documentation's Secure the Activation of Traffic Traces page (https://goo.gl/uDk91k).

Keep in mind that increasing the log level will cause the log and trace files to grow faster. Consider doing so on a case-by-case basis and decreasing the log level again once you are done debugging.

6.3 Backing Up and Restoring Your Configuration

To back up your current Cloud Connector configuration, store a copy of the following three directories:

- *scc_config*
- *config*
- *config_master*

All three directories are located in the Cloud Connector installation directory.

Note

If you are running the Linux installer version, the configuration directories can be found in the */opt/sap/scc* directory.

To restore such a backup, replace the three current configuration directories of your installation with the backed-up versions and then restart Cloud Connector, as described in Section 2.5.

Warning

If you need to restore backed-up configuration from an older version of Cloud Connector to a newer version, SAP recommends that you only do so within the same minor version. This means that you should only restore configuration files from, for example, version 2.10.x to version 2.10.y. The reason for this restriction is that the format of the configuration files might change across minor and major versions, but never within the same minor version.

6.4 Cloud Connector APIs

Cloud Connector offers four APIs that let you query the state of the Cloud Connector instance. Using these APIs, you can, for instance, configure your monitoring solution to monitor Cloud Connector or write ad hoc scripts and programs to perform monitoring or reporting. The APIs are called over HTTPS, and apart from the health check API, which generates no output, they return data in JSON format to the caller.

To access the APIs, the caller must authenticate using HTTP Basic Authentication. If LDAP authentication is configured, the user must be assigned at least one of the roles `sccmonitoring` and `sccadmin`. If LDAP authentication is not configured, the caller must authenticate as the administrator user. For more information on LDAP authentication, please see Section 7.5.

Note

LDAP authentication is the only way to create additional users that are only authorized to call the Cloud Connector APIs. To do so, create the users in the configured directory and assign them the `sccmonitoring` role.

If you want to try out the APIs in the browser, HTTP Basic Authentication is not needed. Simply go to the URL in your preferred browser, and you will be redirected to the login screen for authentication.

In the following sections, you will be introduced to the four APIs and see examples of their output. For a detailed description of the returned JSON data, please refer to the documentation's Monitoring APIs page (https://goo.gl/UFVyEk).

Health Check

The health check API, introduced in Cloud Connector version 2.10.0, lets you determine whether or not a Cloud Connector instance is up and running at a specified host and port. The API is available at the following URL:

https://<host>:<port>/exposed?action=ping

The health check API does not return any data, but if a Cloud Connector instance is running on the provided host and port, the call will return HTTP status code 200.

Note

An HTTP status code of 200 *only* tells you that a Cloud Connector instance is up and running. If, for instance, a network problem has caused the connection to SAP Cloud Platform to break at the time of the API call, the health check API will still return HTTP status code 200.

List of Subaccounts

The list of subaccounts API, introduced in Cloud Connector version 2.10.0, returns a list of the connected SAP Cloud Platform subaccounts, along with information about each subaccount such as region host, subaccount name, state of the tunnel connection, etc. The API is available at the following URL:

https://<host>:<port>/api/monitoring/subaccounts

An example of the JSON data returned by the list of subaccounts API is shown in Figure 71.


```
{
  "subaccounts": [
    {
      "tunnel": {
        "state": "Connected",
        "connectedSince": "2017-08-13T08:03:46.880 +0200",
        "connections": 1,
        "applicationConnections": [
          {
            "connectionCount": 1,
            "name": "services:dispatcher",
            "type": "JAVA"
          }
        ],
        "serviceChannels": []
      },
      "regionHost": "hanatrial.ondemand.com",
      "subaccount": "[redacted]",
      "locationID": ""
    }
  ],
  "version": 1
}
```

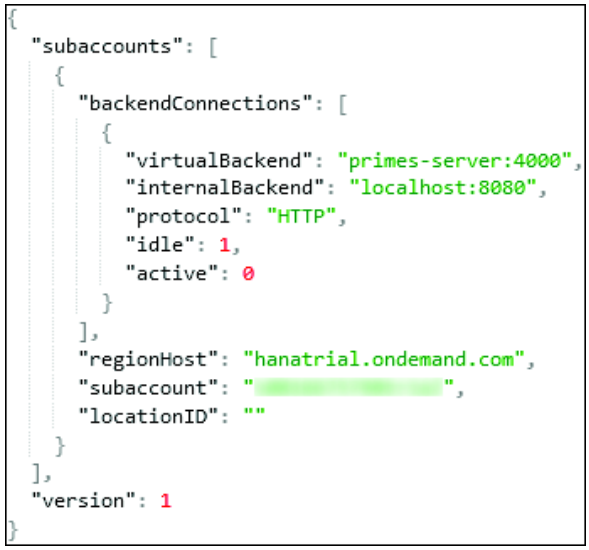

Figure 71 The JSON Output of the List of Subaccounts API

List of Open Connections

The list of open connections API, introduced in Cloud Connector version 2.10.0, returns information about the number of active and idle connections to each configured backend system for each connected subaccount. The API is available at the following URL:

https://<host>:<port>/api/monitoring/connections/backends

An example of the JSON data returned by the list of open connections API is shown in Figure 72.



```
{
  "subaccounts": [
    {
      "backendConnections": [
        {
          "virtualBackend": "primes-server:4000",
          "internalBackend": "localhost:8080",
          "protocol": "HTTP",
          "idle": 1,
          "active": 0
        }
      ],
      "regionHost": "hanatrial.ondemand.com",
      "subaccount": "[redacted]",
      "locationID": ""
    }
  ],
  "version": 1
}
```


Figure 72 The JSON Output of the List of Open Connections API

Performance Monitor Data

The performance monitor data API, introduced in Cloud Connector version 2.10.0, returns information about the amount of traffic going through Cloud Connector to each configured backend system for each connected subaccount. The API is available at the following URL:

https://<host>:<port>/api/monitoring/performance/backends

Figure 73 shows an excerpt of the data returned by the performance monitor data API. The example shows that since monitoring began, three HTTP calls were made to the *primes-server* virtual host and that the calls took between 0 and 10 milliseconds to complete.

```
{
  "subaccounts": [
    {
      "backendPerformance": [
        {
          "virtualHost": "primes-server",
          "virtualPort": "4000",
          "protocol": "HTTP",
          "buckets": [
            {
              "numberOfCalls": 3,
              "minimumCallDurationMs": 0
            },
            {
              "numberOfCalls": 0,
              "minimumCallDurationMs": 10
            },
            {
              "numberOfCalls": 0,
              "minimumCallDurationMs": 20
            },
```


Figure 73 Excerpt of the JSON Data Returned by the Performance Monitor Data API

6.5 Alerting

Cloud Connector continuously monitors for service disruptions and circumstances that can potentially cause disruptions in the future. For each detected issue, Cloud Connector generates an alert. The currently active alerts are listed in the **Connector • Alerting** view, as shown in Figure 74.

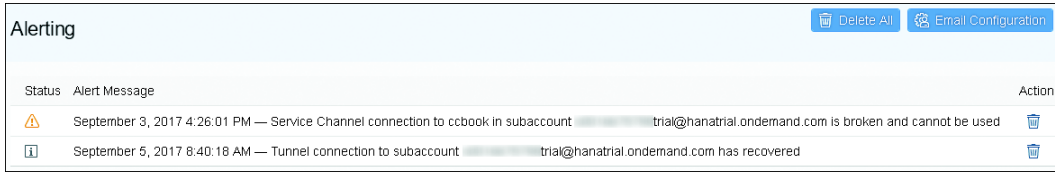


Figure 74 The List of Currently Active Alerts

Cloud Connector checks for issues at fixed intervals. However, an ongoing issue, such as a broken connection to SAP Cloud Platform, will appear only once in the list of active alerts. Once the issue has been resolved, the alert will be removed from the list automatically.

You can manually delete a single alert by clicking its **Delete** icon, or all alerts at once by clicking the **Delete All** button. However, if you delete an alert about an ongoing issue, the alert will be added back to the list when Cloud Connector detects the issue again.

In the two following sections, you will learn how to configure which issues you want to be alerted about and how to set up Cloud Connector to deliver alert messages via email.

Configuring Alert Triggers

To configure what you want to be alerted about, click the **Observation Configuration** button in the **Connector • Alerting** view. This opens the **Observation Configuration** dialog, shown in Figure 75.

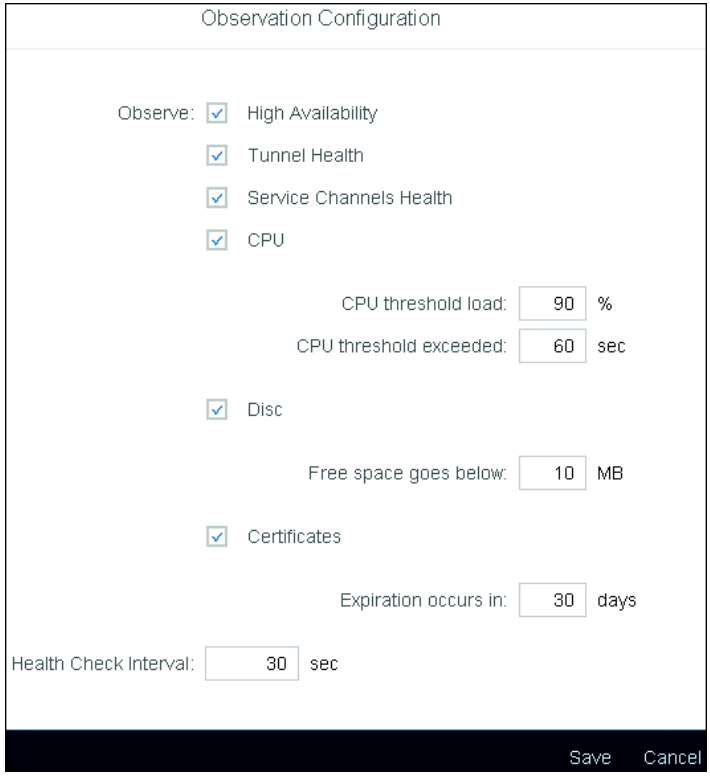


Figure 75 Choosing What You Want to be Alerted About

Select the **High Availability** checkbox if you want to be alerted about issues pertaining to the high availability setup.

To receive alerts about the TLS tunnel that connects Cloud Connector to SAP Cloud Platform, select the **Tunnel Health** checkbox.

You can also choose to receive alerts related to service channels. To do so, select the **Service Channels Health** checkbox.

If you want to be alerted about exceedingly high CPU usage on the Cloud Connector host, select the **CPU** checkbox. You can specify the CPU load alert threshold in the **CPU threshold load** field. The default value is 90%. The **CPU threshold exceeded** field contains the number of seconds the threshold must be exceeded before an alert is triggered. The default value is 60 seconds.

To be alerted about low remaining disk space, select the **Disc** checkbox. You can specify the remaining disk space threshold in the **Free space goes below** field. The default value is 10 MB.

To be alerted about the impending expiration of a certificate used by Cloud Connector, select the **Certificates** checkbox. You can specify when to be alerted in the **Expiration occurs in** field. The default setting is 30 days before the certificate expires.

You can also configure how often Cloud Connector should perform the selected checks. The default value is every 30 seconds. To specify another interval, enter it into the **Health Check Interval** field.

When you are happy with the alert configuration, click **Save** to store it and close the dialog.

Setting Up Email Alerts

To receive alerts via email, you need to provide Cloud Connector with email configuration information such as the receiver email addresses and the SMTP server to employ for sending out the alert messages. To set this up, go

to the **Connector • Alerting** view and click the **Email Configuration** button. This opens the **Email Configuration** dialog, shown in Figure 76.

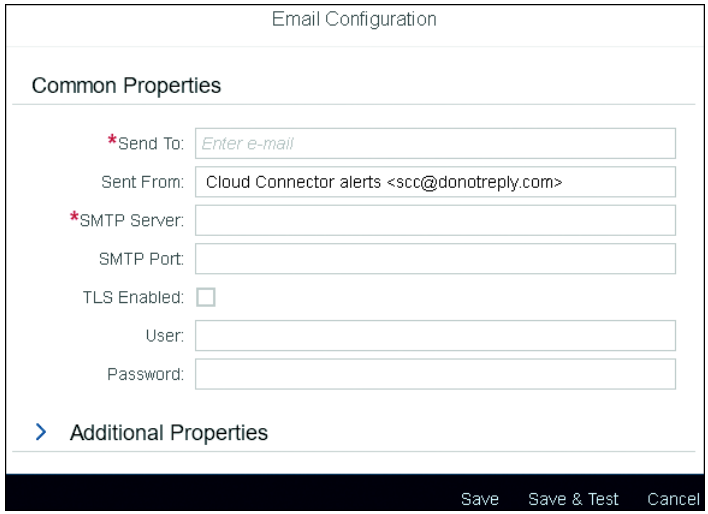


Figure 76 Providing the Configuration Required for Alerts via Email

The **Send To** field is mandatory. It contains the email addresses that are to receive the alert messages. To specify multiple receivers, press Enter after each email address.

The **Sent From** field contains the sender of the alert emails.

The mandatory **SMTP Server** field contains the address of the SMTP server that will be sending out the alert messages. The server must be reachable from the Cloud Connector host.

The **SMTP Port** field contains the port number of the SMTP server. This field is optional. If you do not specify a port number, the underlying JavaMail API framework will use a default value.

If the specified SMTP server supports TLS security, select the **TLS Enabled** checkbox. If not, clear the checkbox.

If the SMTP server requires authentication to send out emails, a user name and password must be provided in the **User** and **Password** fields, respectively.

For advanced configuration, you can pass additional properties directly to the underlying JavaMail API framework. To do so, expand the **Additional Properties** section, shown in Figure 77.

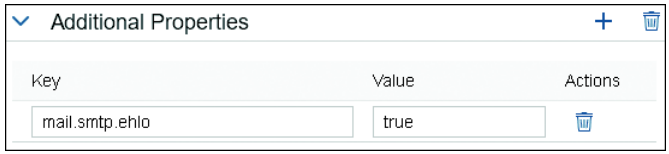


Figure 77 The Additional Properties Section of the Email Configuration Dialog

To add a property, click the **Add a property** icon + and enter the property's name and value into the **Key** and **Value** fields, respectively. To remove a property, click the **Delete** icon 🗑 next to it. You can also remove all properties by clicking the **Delete all properties** icon 🗑.

Note

For a list of supported properties, please refer to the JavaMail documentation (https://goo.gl/WeLY3M).

When you have filled out the fields of the **Email Configuration** dialog, click either **Save** to store the email configuration or **Save & Test** to store the configuration and validate the settings by sending a test email.

6.6 Configuring High Availability

For your on-premise resources to be available to your SAP Cloud Platform applications and services, a Cloud Connector instance must be up and running and connected to the SAP Cloud Platform subaccount. This means that disruptions of the Cloud Connector host, for instance, a hardware failure or

a planned operating system upgrade, will render your on-premise resources inaccessible from the cloud for the duration of the disruption. In a production scenario, this is not acceptable.

To solve this problem, Cloud Connector supports a high availability setup, based on the concept of *master* and *shadow* instances. The master instance is the instance type you have been working with up until this point. A master instance connects to one or more SAP Cloud Platform subaccounts, lets you configure access to on-premise resources, and so on.

A shadow instance, on the other hand, connects to a master instance and receives the master instance's configuration. When the configuration changes on the master instance, the new configuration is automatically pushed to the connected shadow instance.

The shadow instance continually monitors the availability of the master instance. Should the master instance become unavailable, the shadow instance will try to become a master instance and establish connections to the previously configured SAP Cloud Platform subaccounts, thereby making the on-premise resources available again.

When the former master instance comes back online, it will automatically discover that a new master instance has taken its role. It will then proceed to connect to the new master instance, becoming a shadow instance in the process.

A master instance can have at most one shadow instance connected to it, and a shadow instance can be connected to at most one master instance. A shadow instance that isn't connected to a master instance, however, doesn't do anything.

The shadow instance UI is limited, since most configuration is done on the master instance. On-premise resources, subaccount connections, etc. can be viewed in the shadow instance UI but can only be edited on the master instance.

In the following sections, we cover how to enable high availability, how to install a shadow instance, and how to connect it to a master instance.

Enabling High Availability

The first step of configuring the high availability setup is to enable high availability on the master instance. To do so, go to the **Connector • High Availability** view and click the **Enable** button. Figure 78 shows the **High Availability** view with high availability enabled but no shadow instance connected.

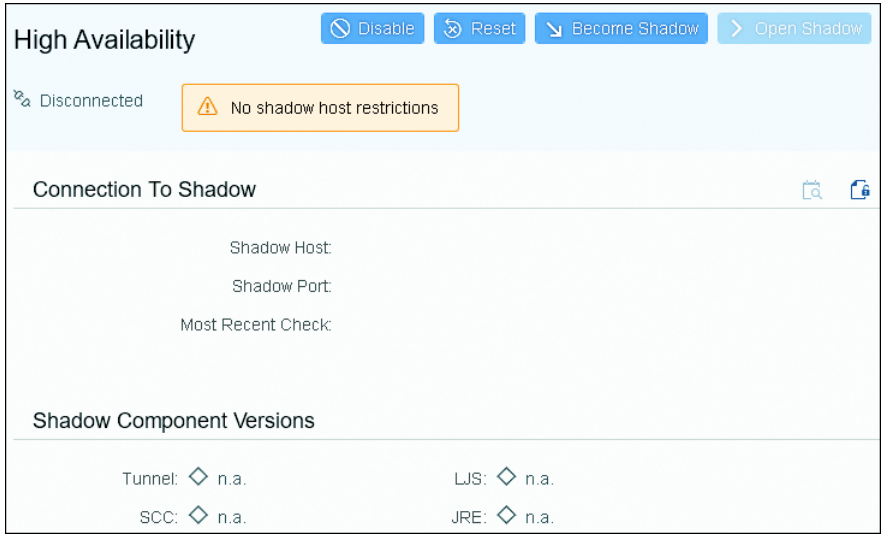


Figure 78 The High Availability View Immediately after Enabling High Availability

The "No shadow host restrictions" message refers to the fact that you can optionally specify a shadow instance host name. If you do so, only a shadow instance running on a host with the specified name will be allowed to connect to this master instance. To specify a shadow instance host name, click the **Set or remove shadow host restriction** icon to open the **Edit High Availability Settings** dialog, shown in Figure 79.

Enter a host name into the **Shadow Host** field, or clear the field if you do not wish to specify a shadow instance host name. Click **Save** to store the configuration and close the dialog. With the setting shown in Figure 79, only a shadow instance running on a host named *shadow-instance-host* will be able to connect to this master instance.

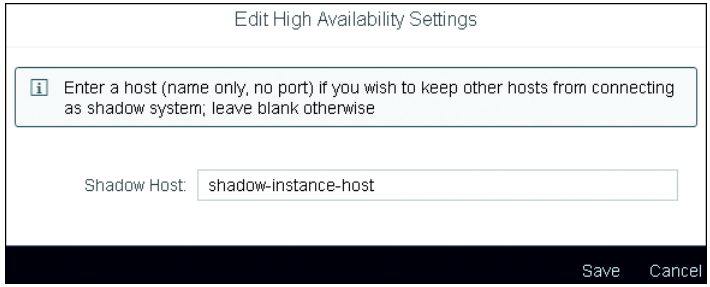


Figure 79 Setting a Shadow Instance Host Name Restriction

Installing and Connecting the Shadow Instance

To install a shadow instance, follow the instructions given in Section 2 until you reach the **Initial Setup** screen. Set a new password for the `Administrator` user, then change the **Choose Installation Type** setting to **Shadow (Backup Installation)**. Click the **Save** button to store the settings, and you will be taken to the **Shadow Connector** view, shown in Figure 80.

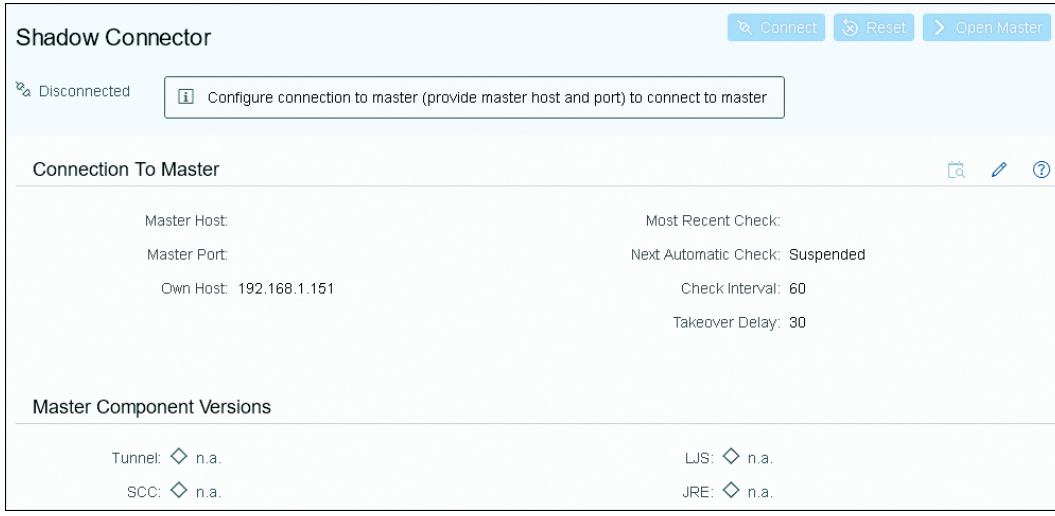


Figure 80 A Freshly Installed and Still Unconnected Shadow Instance

Before the shadow instance can be connected to the master instance, you need to provide the required connection information. To do so, click the **Edit** icon in the **Connection to Master** section to open the **Edit Connection To Master** dialog, shown in Figure 81.

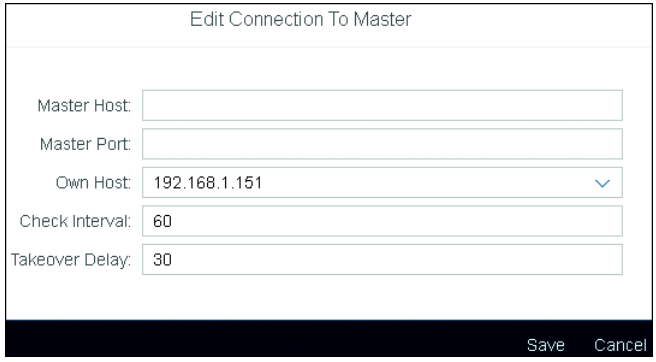


Figure 81 Editing the Master Instance Connection Information

The **Master Host** and **Master Port** fields contain the host name and port, respectively, of the master instance you wish to connect to.

The **Own Host** drop-down list contains the host names associated with the host running the shadow instance. The host name you choose will be passed to the master instance, which will in turn use it to connect to the shadow instance. It is therefore important that you choose a host name which the master instance is able to resolve. If the correct host name is not in the list, you can type it into the field.

The **Check Interval** field contains the time interval in seconds between the shadow instance's checks of the availability of the master instance. The default value is 60 seconds.

The **Takeover Delay** field contains the number of seconds the shadow instance will wait before trying to become a master instance after having discovered that its master instance is no longer available. The default value is 30 seconds.

Note

The shadow instance does not take over immediately after the master instance has become unavailable. The shadow instance first has to detect the unavailable master instance, then wait for a configurable period of time, before trying to take the master role. The time between the master instance becoming unavailable and the shadow instance trying to take the master role can therefore be up to the sum of the values of the **Check Interval** and **Takeover Delay** fields. During this period of time, the configured on-premise and cloud resources will not be available.

Fill out the **Master Host** and **Master Port** fields, and adjust the default values of the **Own Host**, **Check Interval**, and **Takeover Delay** fields if required. Then click **Save** to store the settings and close the dialog.

Note

For the high availability setup to work, the master instance and the shadow instance must be able to reach each other over the network.

With the connection information in place, connect to the master instance by clicking the **Connect** button. Once the connection has been established, the **Shadow Connector** view will be updated with the connection's details, as shown in Figure 82.

If you go to the **Connector • High Availability** view on the master instance, you will notice that it has been updated with the details of the connected shadow instance, as shown in Figure 83.

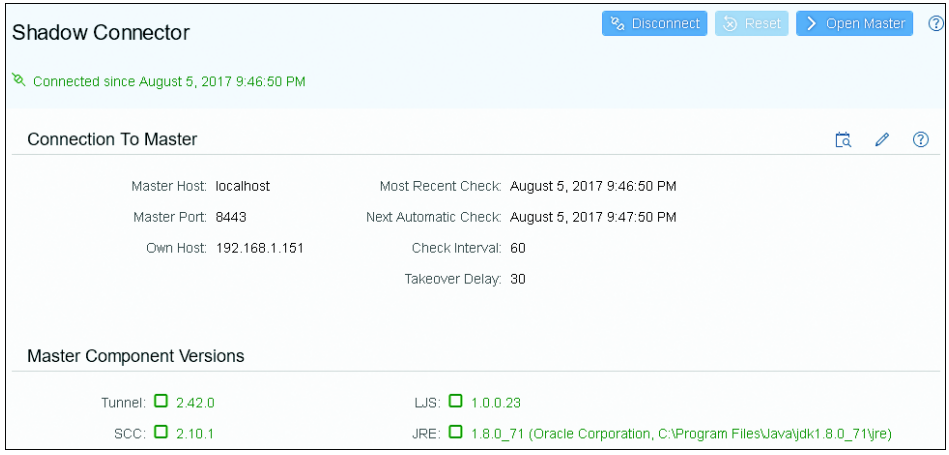


Figure 82 The Shadow Instance Is Now Connected to Its Master Instance

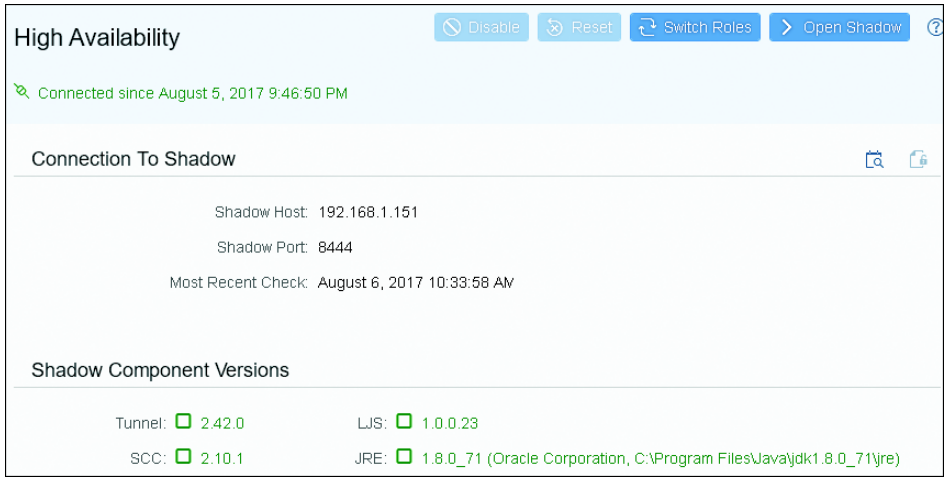


Figure 83 The High Availability View Displays the Details of the Connected Shadow Instance

The **Connectivity • Cloud Connectors** view in the SAP Cloud Platform cockpit also reflects that high availability has been enabled and a shadow instance has been connected, as shown in Figure 84.

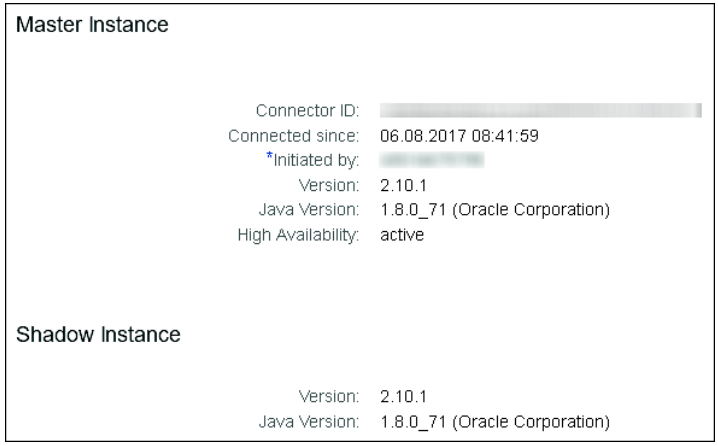


Figure 84 The Cloud Connectors View with High Availability Enabled and a Connected Shadow Instance

If you wish to disconnect a shadow instance from its master instance, you can do so by clicking the **Disconnect** button in the **Shadow Connector** view. You can also reset the high availability configuration on both a shadow and a master instance, provided the two are not connected at the time. To reset a master instance, click the **Reset** button in the **Connector • High Availability** view. To reset a shadow instance, click the **Reset** button in the **Shadow Connector** view.

6.7 Minimizing Downtime during Upgrades

As we discussed in Section 2.7, upgrading a Cloud Connector installation will cause it to be offline for a brief period of time. However, with a high availability setup in place, downtime during upgrades can be minimized.

To perform a near-zero downtime upgrade in a high availability environment, follow the steps outlined below:

1. Stop the current shadow instance.
2. Upgrade the stopped shadow instance.

3. Start the upgraded shadow instance. It will automatically reconnect to the master instance.
4. Switch the roles of the two instances by clicking the **Switch Roles** button in the **Connector • High Availability** view in the master instance UI.
5. Stop the new shadow instance (which was previously the master instance).
6. Upgrade the stopped shadow instance.
7. Start the upgraded shadow instance. It will automatically reconnect to the master instance.
8. Switch the roles of the two instances back by clicking the **Switch Roles** button in the **Connector • High Availability** view in the master instance UI.

When you have completed these eight steps, both instances will have been upgraded to the new version, and both will have the same role in the high availability setup as before the upgrade.

Note

This upgrade procedure results in near-zero downtime, not zero downtime. When the master and shadow roles are switched, there will be a few seconds during which configured resources are not available.

For instructions on starting and stopping Cloud Connector, please see Section 2.5. Section 2.7 describes how to upgrade Cloud Connector.

6.8 Changing the Cloud Connector UI Port Number

The port number that the Cloud Connector UI is available on can be changed at any time by executing a script from the command line. The script is called *changeport.bat* on Microsoft Windows and *changeport.sh* on Linux and macOS, and you can find it in the root of the Cloud Connector installation directory.

Like the Cloud Connector launch scripts, this script must also be able to execute the `java` command. For it to function properly, either your Java installation's *bin* directory must be added to the `PATH` environment variable or the `JAVA_HOME` environment variable must be set, pointing to your Java installation directory.

The script expects a single parameter, which is the new port number you want the UI to be available on. In the following example, the port number is changed to 9000 on Microsoft Windows.

```
cd <your Cloud Connector installation directory>
changeport.bat 9000
```

For the change to take effect, Cloud Connector must be restarted. For a refresher on how to do that, please refer to Section 2.5.

If you are running the installer version on Linux, the approach is slightly different. The `rpm` tool installs the Cloud Connector files into the */opt/sap/scc* directory, and this is where the *changeport.sh* script is located. However, only the `sccadmin` user and its group `sccgroup` have access to the directory and the script. Therefore, we must temporarily switch identity to the root user to execute the script. Listing 3 shows you how to accomplish this.

```
su
cd /opt/sap/scc
./changeport.sh <the new port number>
service scc_daemon restart
exit
```

Listing 3 Changing the Port Number of the Linux Installer Version's UI

7 Security

In previous sections, we covered the hybrid cloud landscape, how to install Cloud Connector, how to configure the cloud to on-premise and on-premise

to cloud scenarios, and Cloud Connector operations. In this section we will cover the security aspects of Cloud Connector.

Note

Cloud Connector is as much about security as it is about connectivity. Therefore, you should make sure that your security department is involved from the very beginning when you are implementing Cloud Connector.

First, you will learn how to get an overview of your Cloud Connector installation's security status. Then, we will discuss which network zone to run Cloud Connector in. You will also learn how to sign Cloud Connector's UI certificate, to get rid of those browser warnings you keep seeing. Next, you will get a high-level overview of Cloud Connector's principal propagation support. We will also discuss the two ways to authenticate when logging on to Cloud Connector's UI and how to configure whitelisting of trusted SAP Cloud Platform applications and services. Finally, we will cover the audit log and Cloud Connector's trust store.

7.1 The Security Status Overview

The **Connector • Security Status** view, shown in Figure 85, is a convenient way to get an overview of how well your Cloud Connector installation is doing in a number of security area.

The **General Security Status** section lists the security areas that are independent of the connected SAP Cloud Platform subaccounts. Via the icons in the **Actions** column, you can navigate directly to the specific part of the Cloud Connector UI, where you can fix a reported issue.

The **Subaccount-Specific Security Status** section lists the security status in the areas of application whitelisting and payload trace for each connected subaccount.

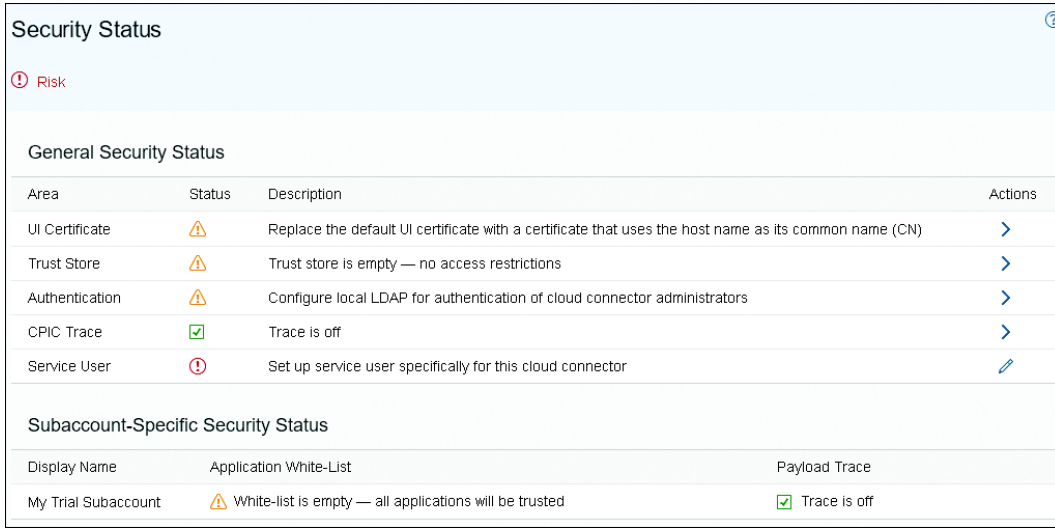


Figure 85 Cloud Connector's Security Status View

The current status of the listed security areas is indicated by an icon. The three statuses are OK ☑, low risk ⚠, and risk ⓘ.

Cloud Connector combines the individual security statuses into a summary, which is displayed at the top of the page, as well as in the **Status Overview** section of the **Connector** view.

All the listed security areas are covered in this E-Bite. Table 4 shows you which sections to go to for more information.

Security area	Section
UI Certificate	Section 7.3
Trust Store	Section 7.8
Authentication	Section 7.4
CPIC Trace	Section 6.2
Service User	Section 7.1
Application White-List	Section 7.6
Payload Trace	Section 6.2

Table 4 Where to Find More Information about the Security Areas

The Microsoft Windows Service User

SAP recommends that the Microsoft Windows installer version's service runs in the security context of a user, which has been created specifically for this purpose and which has limited privileges.

> **Note**
>
> It is also recommended that the Cloud Connector files and directories are made accessible only to this user and the system administrators.

However, Cloud Connector cannot automatically detect if this is the case. Therefore, you need to manually update the security status when the service user has been configured. To do so, click the **Edit** icon in the **Actions** column of the **Service User** row. This opens the **Edit Service User** dialog, shown in Figure 86.

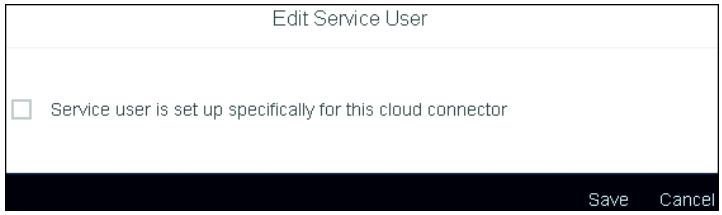


Figure 86 Manually Indicating the Microsoft Windows Service User Security Status

If the Microsoft Windows service user has been configured correctly, select the checkbox and click **Save** to store the information and close the dialog.

> **Note**
>
> The service user security area is displayed only on Microsoft Windows installations.

7.2 Where to Run Cloud Connector

One of the major differences between Cloud Connector and a reverse proxy is that the Cloud Connector host does not need to be—and indeed never

should be—reachable from the Internet. This means that, unlike with a reverse proxy, we have a choice in where to run Cloud Connector: either in the demilitarized zone (DMZ), between the outermost and the innermost firewall, or in the internal network, behind the innermost firewall.

Both scenarios have pros and cons, and both have been implemented by real-world customers. In the next two sections, we discuss both architectures in more detail. As for choosing one over the other, please keep the advice from this section's introduction in mind: make sure to involve your colleagues in the security department.

Running Cloud Connector in the DMZ

In this scenario, Cloud Connector runs on a host in the DMZ, between the outermost and the innermost firewall. Figure 87 shows a diagram of the architecture.

Note

In Figure 87, arrows represent network connections and point *away* from the system initiating the connection.

Cloud Connector opens a connection to SAP Cloud Platform to establish the TLS tunnel. This means that the Cloud Connector host must be able to reach SAP Cloud Platform, either directly or through an HTTPS proxy. In the former case, the outermost firewall's outbound traffic rules must be updated accordingly.

The Cloud Connector host must also be able to reach those backend systems in the internal network whose resources are to be made available to applications in the cloud. This means that the innermost firewall's inbound traffic rules must be updated for every backend system and protocol configured in Cloud Connector.

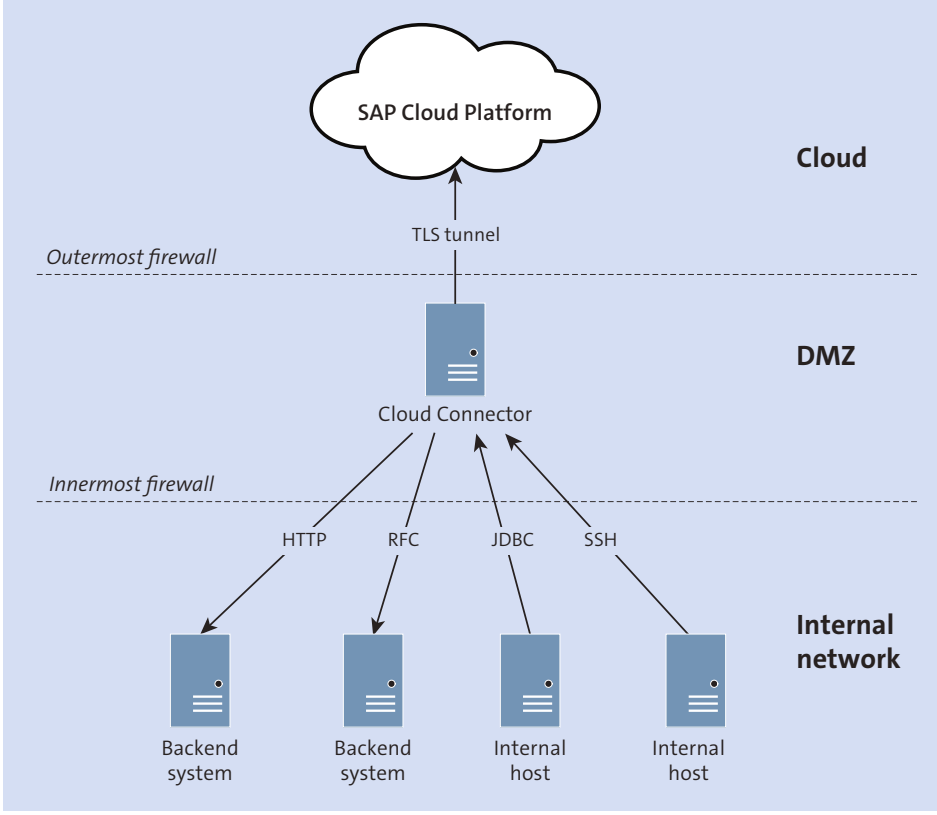


Figure 87 Cloud Connector Running in the DMZ

In addition, hosts in the internal network that need to access resources in the cloud through a Cloud Connector service channel must be able to reach the Cloud Connector host in the DMZ. As a result, the innermost firewall's outbound traffic rules must be updated to allow these connections.

To support this scenario, the security department must work in tandem with developers and project teams. This introduces a dependency, which must be managed. Also, maintaining firewall rules takes ongoing effort. However, this architecture also provides a high level of security. Placing Cloud Connector in the DMZ means that the damage caused by, say, a software exploit or a rogue employee can be contained.

When considering this architecture, you must weigh the ongoing effort required to maintain it and the increased dependency on the security department against the high level of security the architecture provides.

Note

Some corporate networks have more than one DMZ. Your network security colleagues will help you determine where to run Cloud Connector in your particular network architecture.

Running Cloud Connector in the Internal Network

In this scenario, Cloud Connector runs on a host in the internal network, behind the innermost firewall. Figure 88 shows a diagram of the architecture.

Note

In Figure 88, arrows represent network connections and point *away* from the system initiating the connection.

Like in the previous scenario, Cloud Connector opens a connection to SAP Cloud Platform to establish the TLS tunnel. This requires that the Cloud Connector host can reach SAP Cloud Platform, either directly or through an HTTPS proxy. In the former case, the outbound traffic rules of both firewalls must be updated to allow this connection.

This, however, is the *only* update to firewall traffic rules that's needed. The Cloud Connector host is located in the same network zone as both the backend systems, whose resources are to be made available to cloud applications, and the internal hosts, which need to access cloud resources through Cloud Connector service channels. Therefore, network connections between them do not pass through any firewalls.

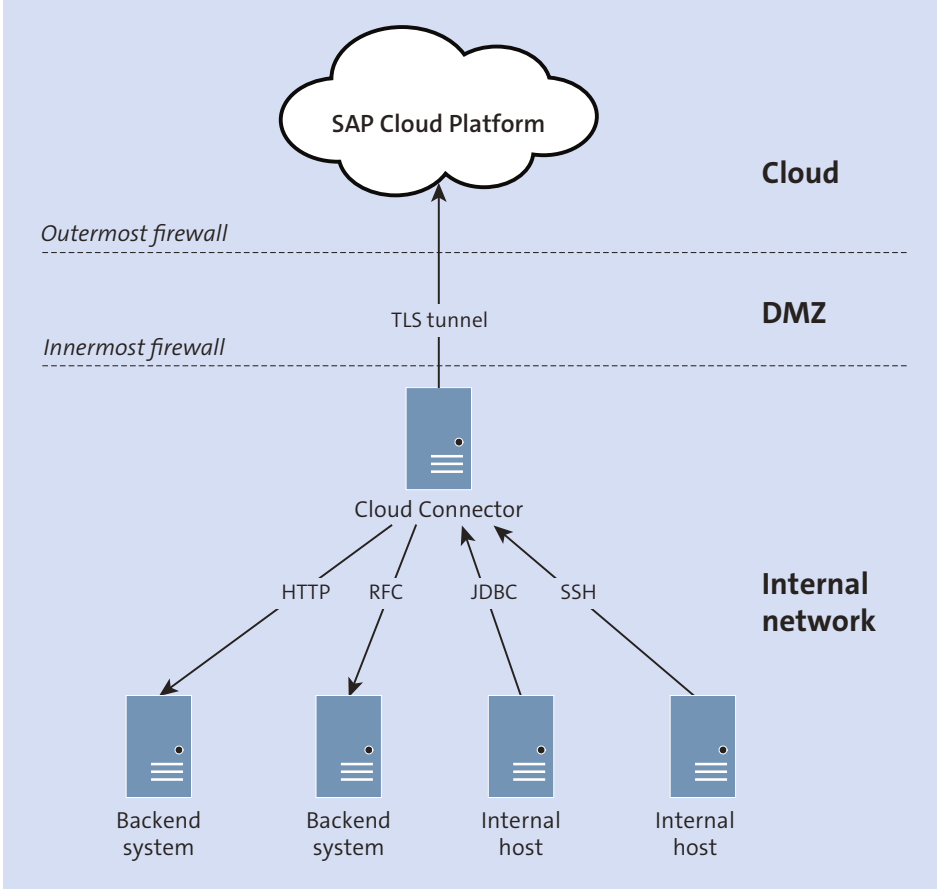


Figure 88 Cloud Connector Running in the Internal Network

Additionally, developers and project teams are less dependent on the security department in this scenario, since ongoing updates to firewall traffic rules are not needed. This lets you use Cloud Connector in a more agile fashion.

But as appealing as the benefits of this architecture are, it does have one major drawback: it offers less security than running Cloud Connector in the DMZ. Since there is no direct control over what Cloud Connector can access on the internal network, the damage of a security incident, should one occur, is potentially far greater.

When considering this architecture, you must weigh the increased agility and reduced maintenance effort against the decrease in security level.

7.3 Signing the UI Certificate

Cloud Connector ships with a self-signed certificate, which is used when a browser accesses the Cloud Connector UI over HTTPS. Since the certificate isn't signed by a trusted certificate authority (CA), browsers will issue a security warning when they encounter it, similar to the one shown in Figure 89.

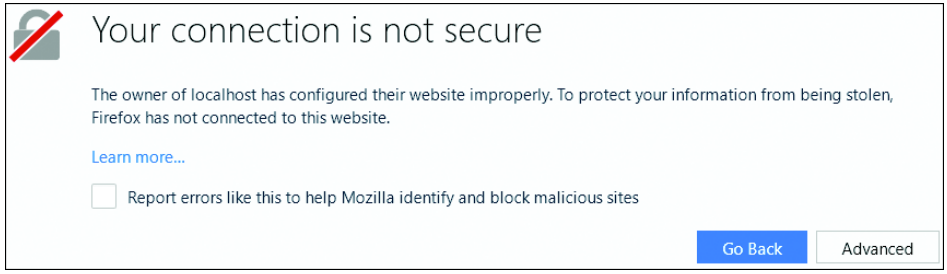


Figure 89 The Firefox Browser Warns about an Untrusted Certificate

To get rid of the browser warnings, Cloud Connector's self-signed certificate must be replaced with a certificate which has been signed by a CA. To do that, you can either create a certificate signing request, have it signed by a CA, and import the resulting certificate, or you can reuse Cloud Connector's system certificate. Both approaches are covered in the following sections.

Create a Signing Request

To sign Cloud Connector's certificate, you need to create a certificate signing request (CSR), which is then processed by a CA. This results in a signed certificate, which can be imported into Cloud Connector.

To generate a CSR, go to the **USER INTERFACE** tab of the **Connector • Configuration** view. In the **UI Certificate** section, click the **Generate a certificate signing request** icon to open the **Generate CSR** dialog, shown in Figure 90.

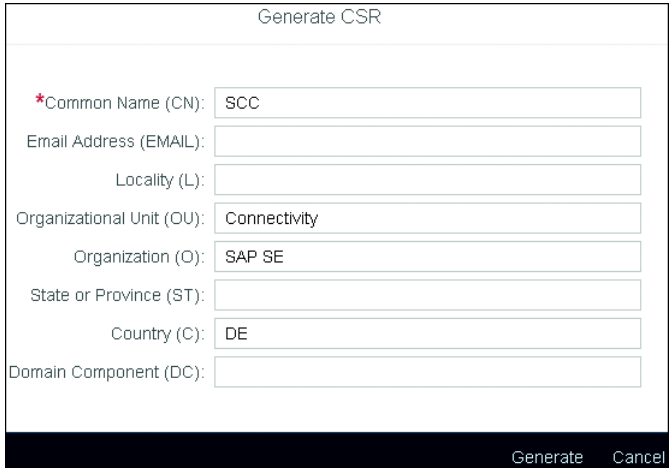


Figure 90 Generating a Certificate Signing Request

The common name attribute, which you enter into the **Common Name (CN)** field, must match the fully qualified domain name of the Cloud Connector host.

Your company's security department will be able to assist you with the appropriate values for the remaining attributes of the CSR.

Click **Generate** to create the CSR in the PEM format, and then save the resulting file. Once created, the CSR must be handed over to the CA for signing. The specifics of this process depend on your company's chosen public key infrastructure (PKI), but the outcome is always a signed certificate, which must be imported into Cloud Connector.

Note

Since its version 58, the Google Chrome browser uses the Subject Alternative Name (SAN) X.509 extension, not Common Name, to match the fully qualified domain name of the host. Therefore, to avoid warnings in Google Chrome, the CA must add this extension to the signed certificate.

To import the signed certificate, click the **Import a certificate** icon in the **UI Certificate** section. This opens the **Import UI Certificate** dialog, shown in Figure 91.

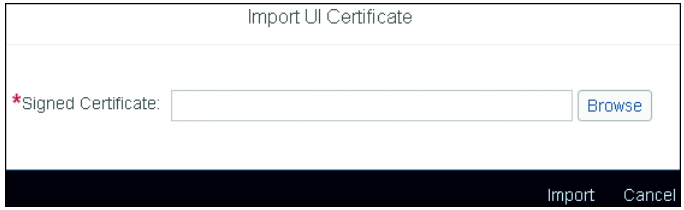


Figure 91 Importing a Signed UI Certificate

Click the **Browse** button, and select the certificate file from its location. Then click **Import** to import the signed certificate and close the dialog.

> **Note**
>
> The imported certificate must be the result of the CA signing the certificate signing request. Importing another signed certificate will fail.

To activate the new UI certificate, Cloud Connector must be restarted. For a refresher on how to do that, please see Section 2.5.

Copy the System Certificate

If a signed system certificate (i.e., the certificate that identifies this particular installation) has been imported into Cloud Connector, you can reuse that certificate as a UI certificate. To do so, click the copy icon in the **UI Certificate** section. A dialog will pop up, informing you that the current UI certificate will be overwritten and asking you to confirm. Click **OK** if you want to go ahead with copying the system certificate, or otherwise click **Cancel**.

After the system certificate has been copied, Cloud Connector must be restarted for the change to take effect. For more information on how to restart Cloud Connector, please see Section 2.5.

Note

Your security department might have a policy against using the same certificate for multiple purposes, so consult with them before reusing the system certificate as a UI certificate.

Excluding Cipher Suites

When a browser visits the Cloud Connector UI, a TLS handshake takes place to establish secure communications. During this handshake, the browser and Cloud Connector agree on how the connection should be encrypted. In TLS terms, they choose a *cipher suite*.

By default, Cloud Connector will agree to use any cipher suite supported by the underlying Java Virtual Machine. However, corporate security policies might dictate that certain cipher suites should not be used, because they are deemed insecure. To accommodate this scenario, Cloud Connector lets you exclude those cipher suites from being used in TLS communications.

To exclude one or more cipher suites, go to the **USER INTERFACE** tab of the **Connector • Configuration** view. The **Cipher Suites** section lists the cipher suites that are eligible to be used in TLS communications. To update the list, click the **Edit** icon . This opens the **Edit Eligibility Of Cipher Suites** dialog, shown in Figure 92.

To exclude a cipher suite, select its checkbox on the left-hand side of the dialog, and click the arrow pointing right →. To add a cipher suite back to the list of eligible cipher suites, select its checkbox on the right-hand side of the dialog, and click the arrow pointing left ←.

When you are finished updating the list of eligible cipher suites, click **Save** to store the configuration and close the dialog.

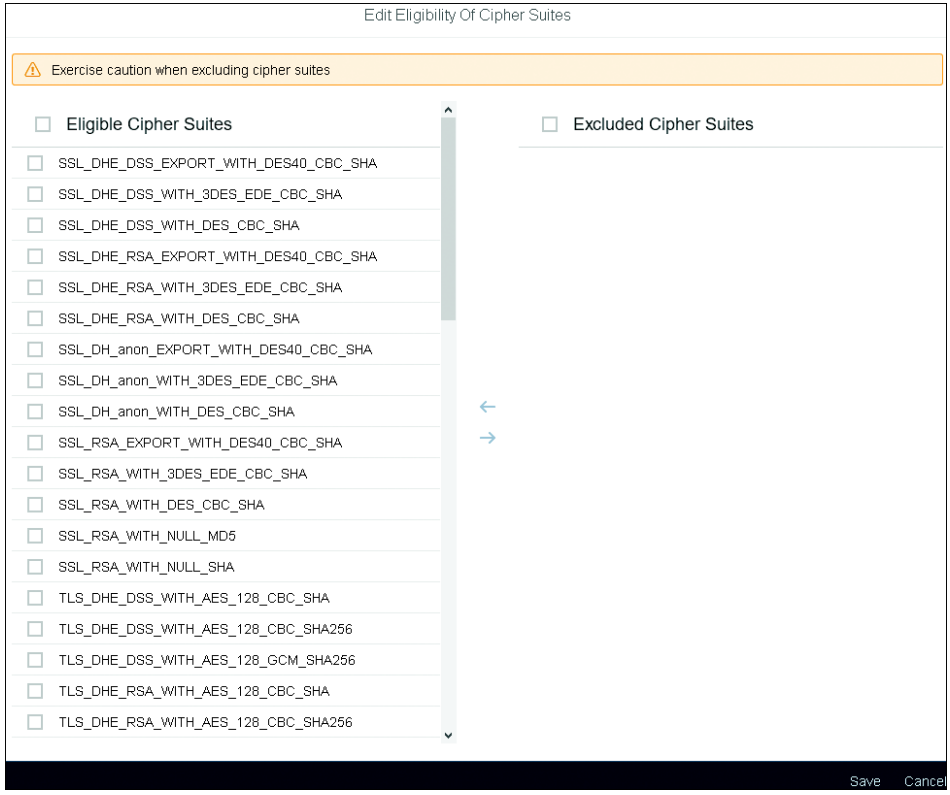


Figure 92 Updating the List of Eligible Cipher Suites

7.4 Principal Propagation

The purpose of principal propagation is to forward an authenticated user's identity from one system to another in a secure fashion. In the context of Cloud Connector, the authenticated user's identity is being forwarded from SAP Cloud Platform to an on-premise backend system via Cloud Connector.

The details of implementing principal propagation in SAP Cloud Platform, Cloud Connector, and the on-premise backend system are outside of the scope of this E-Bite. Therefore, this section is intended to provide only a high-level overview of the steps involved. For more detailed information on how to implement principal propagation, please see the documentation's

Configuring Principal Propagation page (https://goo.gl/fjuPLK) and its subpages.

A prerequisite for principal propagation is that Cloud Connector trusts the cloud application, which forwards the user's identity. User identity can be provided by SAML2 identity providers such as the SAP ID Service, by SAP HANA instances, and by Java applications. By default, Cloud Connector does not trust any SAP Cloud Platform identity provider. Therefore, Cloud Connector must be configured to trust the appropriate identity provider for your principal propagation scenario.

Cloud Connector supports principal propagation for HTTPS and RFC communications. In both cases, the backend system must trust the Cloud Connector installation before it will accept a forwarded identity. In the HTTPS case, this is accomplished by creating a system certificate identifying the Cloud Connector installation and importing that certificate into the backend system. In the RFC case, the ABAP backend system must be configured to trust Cloud Connector's System PSE.

In the RFC scenario, the user's identity is forwarded to the ABAP backend directly through the RFC protocol. In the HTTPS scenario, the identity is forwarded to the backend system by way of a short-lived X.509 certificate issued and signed by either Cloud Connector or SAP Secure Login Server. A prerequisite for this is that the backend system has been configured for certificate-based authentication.

In both the HTTPS and RFC scenarios, the backend system must map the user's cloud identity to a user name. User mapping can be set up manually on a per-user basis, but for large numbers of users, rule-based mapping is preferable if the backend system supports it.

Cloud Connector also supports forwarding user identity using the Kerberos authentication protocol. In this scenario, Cloud Connector communicates with a Kerberos Key Distribution Center to retrieve a token, which is then passed to the backend system as a credential.

7.5 The Two Authentication Modes of Cloud Connector's UI

Cloud Connector supports two ways of authenticating when logging on to the administration UI: password authentication and LDAP authentication.

Password authentication mode is the default, and the one we've been using so far. In this mode, you log on to Cloud Connector using a single administrator user name and password, regardless of the actual number of Cloud Connector administrators. The administrator user name and password are maintained in the Cloud Connector UI.

In LDAP authentication mode, however, the administrator user is disabled and can no longer be maintained in the Cloud Connector UI. Instead, users are authenticated against a directory service using the LDAP protocol. This lets you add any number of Cloud Connector administrators, each with their own named user. You will learn how to both maintain the administrator user in password authentication mode and configure LDAP authentication mode later in this section.

SAP recommends running Cloud Connector in LDAP authentication mode. This ensures that every administrator is uniquely identified for auditing purposes and that a single password is not shared between multiple administrators. An additional benefit over password authentication mode is that a central directory service can ensure strong passwords through the enforcement of corporate password policies.

Note

If you are running Cloud Connector on your local machine for learning purposes, password authentication mode is sufficient.

Maintaining the Administrator User in Password Authentication Mode

In password authentication mode, the administrator user name and password are maintained in the Cloud Connector UI. To do so, go to the **USER**

INTERFACE tab of the **Connector • Configuration** view. In the **Authentication** section, click the **Edit** icon to open the **Edit Authentication** dialog, shown in Figure 93.

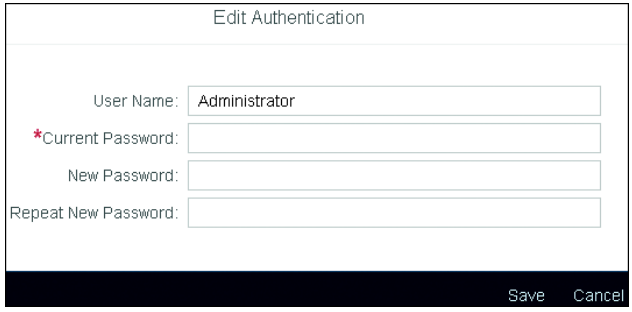


Figure 93 Editing the Administrator User Name and Password

To change the administrator user name, enter the new user name into the **User Name** field and the current password into the **Current Password** field. To update the administrator password, enter the current password into the **Current Password** field and the new password into the **New Password** and **Repeat New Password** fields. Then click **Save** to store the new authentication information.

Note

You cannot update the user name and the password at the same time.

If you lose the administrator password, you will no longer be able to log on to Cloud Connector. It is possible, however, to reset the password to the default value of "manage". How to do this is described in SAP Knowledge Base Article 2388242 (https://launchpad.support.sap.com/#/notes/2388242).

Setting Up LDAP Authentication

To switch from password authentication to LDAP authentication, go to the **USER INTERFACE** tab of the **Connector • Configuration** view. In the **Authentication** section, click the **Switch to LDAP authentication icon** to open the **Edit Authentication** dialog, shown in Figure 94.

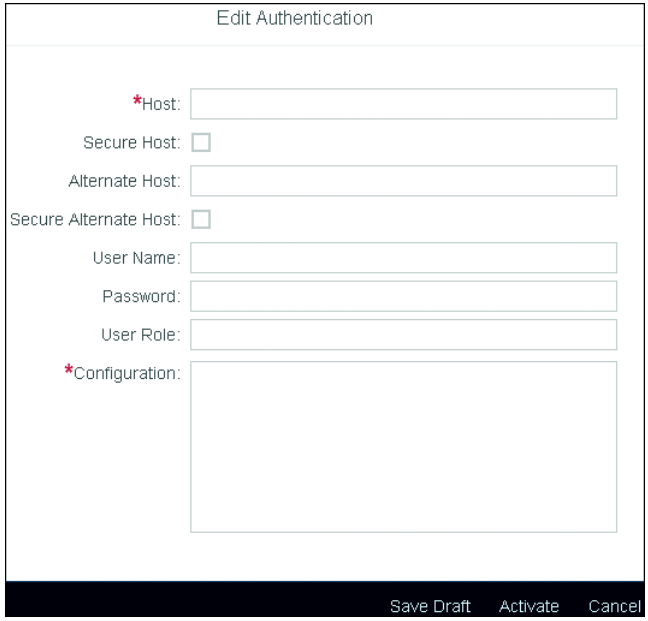


Figure 94 Configuring LDAP Authentication

The **Host** field contains the host name and port number of the directory server, formatted as *host:port*. If the directory server supports LDAP over SSL, select the **Secure Host** checkbox to use it.

If there is a failover directory server in your landscape, you can provide its host name and port number in the optional **Alternate Host** field. If the failover directory server supports LDAP over SSL, select the **Secure Alternate Host** checkbox to use it.

Note

If you provide only a host name, Cloud Connector will use a default port number of 389 for LDAP and 636 for LDAP over SSL.

If you need to provide credentials to the directory server, enter them into the optional **User Name** and **Password** fields.

By default, Cloud Connector requires that a Cloud Connector administrator user has the `sccadmin` role in the directory. If you want to use a different role for authorization, enter the name of that role into the optional **User Role** field.

The mandatory **Configuration** field contains the information needed to locate a user's entry in the directory and to check whether that user has the required Cloud Connector administrator role. For more detailed information on the contents of this field, please see the next section.

When you have provided the LDAP configuration information, you can either store the configuration without activating it or activate it immediately. To do the former, click **Save Draft**. To do the latter, click **Activate**.

Activating LDAP authentication will restart Cloud Connector and end your current session. When you log in again, you must do so with a user from the directory that has the proper Cloud Connector administrator role.

After the switch to LDAP authentication, the **Authentication** section displays the directory server configuration details, as shown in Figure 95.

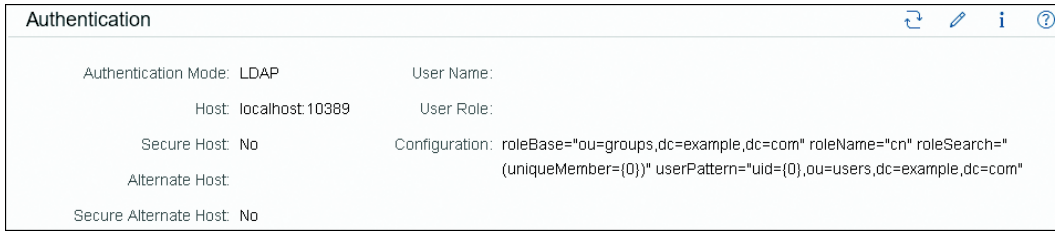


Figure 95 The Directory Server Configuration Details in LDAP Authentication Mode

LDAP Configuration Attributes

For LDAP authentication to function, Cloud Connector needs to know where to locate users and roles in the directory's hierarchy of entries. This information is provided in the LDAP configuration in the form of attribute name-value pairs. Listing 4 contains an example of a complete LDAP configuration.

```
userPattern="uid={0},ou=users,dc=example,dc=com"
roleBase="ou=groups,dc=example,dc=com"
roleName="cn"
roleSearch="(uniqueMember={0})"
```

Listing 4 Sample LDAP Configuration

The `userPattern` attribute describes how to locate users in the directory. In the sample configuration, users are identified by their `uid` attribute and located in the `users` organizational unit, which is located under the `example` and `com` domain components.

The `roleBase` attribute describes how to locate roles in the directory. In the sample configuration, roles are located in the `groups` organizational unit, located under the same node as the users.

The `roleName` attribute contains the name of the attribute that identifies a role. In the sample configuration, roles are identified by their `cn` attribute, i.e., by their common name.

The `roleSearch` attribute describes how to check whether a user has a particular role. In the sample configuration, this is done by checking whether the role has a `uniqueMember` attribute whose value matches the user's unique name.

Note

The LDAP configuration parameters are described in detail in the Apache Tomcat JNDIRealm documentation (https://goo.gl/BhxCgZ).

Keep in mind that the configuration for your particular directory will differ from the sample configuration shown in this section. Your directory service administrator will be able to help you with the configuration details.

Also note that if the LDAP configuration is not correct, you will not be able to log in again after switching to LDAP authentication mode. Should that happen, you need to manually switch back to password authentication. The process for doing so is covered in the next section.

Switching from LDAP Authentication to Password Authentication

You can switch back to password authentication from within the Cloud Connector UI. To do so, click the **Switch to password authentication** icon in the **Authentication** section. However, if you are not able to log in due to an incorrect LDAP configuration, a different approach is needed.

If your Cloud Connector version is 2.8.0 or newer, you can switch to password authentication mode by running a script from the command line. On Microsoft Windows, run the following command from the root of the installation directory:

```
useFileUserStore.bat
```

On Linux and macOS, run the following command, also from the root of the installation directory:

```
./useFileUserStore.sh
```

Note

Both scripts need to be able to execute the `java` command. For the scripts to function, either your Java installation's *bin* directory must be added to the `PATH` environment variable or the `JAVA_HOME` environment variable must be set, pointing to your Java installation directory.

After running the script, restart Cloud Connector to activate the change. How to do this is described in Section 2.5.

If your Cloud Connector version is older than 2.8.0, you need to manually update the *default-server.xml* configuration file. On Microsoft Windows installations, the file is located here:

<installation_directory>\config_master\org.eclipse.gemini.web.tomcat\default-server.xml

On Linux and macOS installations of the portable version, the file is located here:

<installation_directory>/config_master/org.eclipse.gemini.web.tomcat/default-server.xml

On Linux installations of the installer version, the file is located here:

/opt/sap/scc/config_master/org.eclipse.gemini.web.tomcat/default-server.xml

In the *default-server.xml* file, locate the `Realm` element and replace it with the XML fragment in Listing 5.

```
<Realm className="org.apache.catalina.realm.LockOutRealm">
  <Realm className="org.apache.catalina.realm.CombinedRealm">
    <Realm X509UsernameRetrieverClassName=
      "com.sap.scc.tomcat.utils.SccX509SubjectDnRetriever"
      className="org.apache.catalina.realm.UserDatabaseRealm"
      digest="SHA-256" resourceName="UserDatabase"/>
    <Realm X509UsernameRetrieverClassName=
      "com.sap.scc.tomcat.utils.SccX509SubjectDnRetriever"
      className="org.apache.catalina.realm.UserDatabaseRealm"
      digest="SHA-1" resourceName="UserDatabase"/>
  </Realm>
</Realm>
```

Listing 5 Replacement for the Realm Element

After updating the file, restart Cloud Connector for the change to take effect. The restart process is described in Section 2.5.

7.6 Application Whitelisting

In previous sections we covered how Cloud Connector uses whitelisting to, for instance, control access to backend systems and their resources. In this section, you will learn about another security area in which this concept is

applied: restricting the cloud applications that are allowed to access Cloud Connector.

Every connected SAP Cloud Platform subaccount has its own whitelist of trusted applications. Initially, the list is empty. This means that every application and service of the subaccount will be allowed to access Cloud Connector and its configured backend systems and resources.

As soon as an application is added to the whitelist, however, every application *not* on the list will no longer be allowed to access Cloud Connector. When an application is denied access in this way, an entry of type `Denied` will be written to Cloud Connector's audit log. For more information on the audit log, please see Section 7.7.

To view the application whitelist, shown in Figure 96, go to the **TRUSTED APPLICATIONS** tab of the **Subaccount • Cloud To On-Premise** view.

Figure 96 The List of Trusted Applications

To add an application to the list, click the **Add** icon + to open the **Add Trusted Applications** dialog, shown in Figure 97.

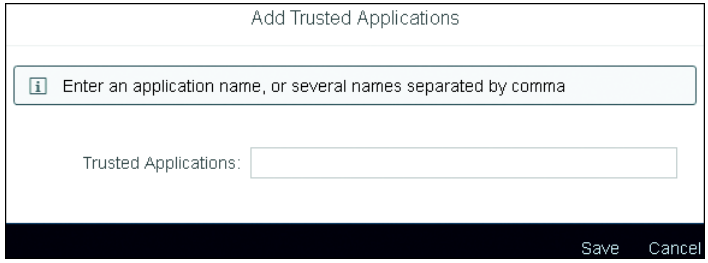


Figure 97 Adding Trusted Applications to the Application Whitelist

Enter one or more application names, separated by commas, into the **Trusted Applications** field. Then click **Save** to add the applications to the whitelist and close the dialog.

An application on the list can be edited by clicking its **Edit** icon and deleted from the list by clicking its **Delete** icon. You can remove every trusted application by clicking the list's **Delete all trusted applications** icon .

Java and SAP HANA XS applications can be whitelisted individually, using the application name displayed in the SAP Cloud Platform cockpit. HTML5 applications, unfortunately, cannot be whitelisted on a per-application basis. Instead, you need to whitelist the `services:dispatcher` application. This will allow every HTML5 application running in the SAP Cloud Platform subaccount to access Cloud Connector.

To whitelist a subscribed application, such as an SAP Cloud Platform service you've activated in your subaccount, format the application name as follows:

```
<provider-subaccount>:<application-name>
```

If you are in doubt as to the correct application name to whitelist, go through the following four steps:

1. If the list of trusted applications is empty, add a dummy application name to it to activate whitelisting.
2. Attempt and fail to access a backend resource through Cloud Connector from the application in question.
3. Retrieve the application name from the audit log by looking for an entry of type `Denied` with the text "Denying access to any system for application <application name here>".
4. Add the application name to the whitelist and delete the dummy application name if you added one.

7.7 The Cloud Connector Audit Log

Cloud Connector's audit log lets you closely track configuration changes performed by administrators, as well as requests for access to backend resources from SAP Cloud Platform. The audit log is available in the **Subaccount • Audits** view, shown in Figure 98.

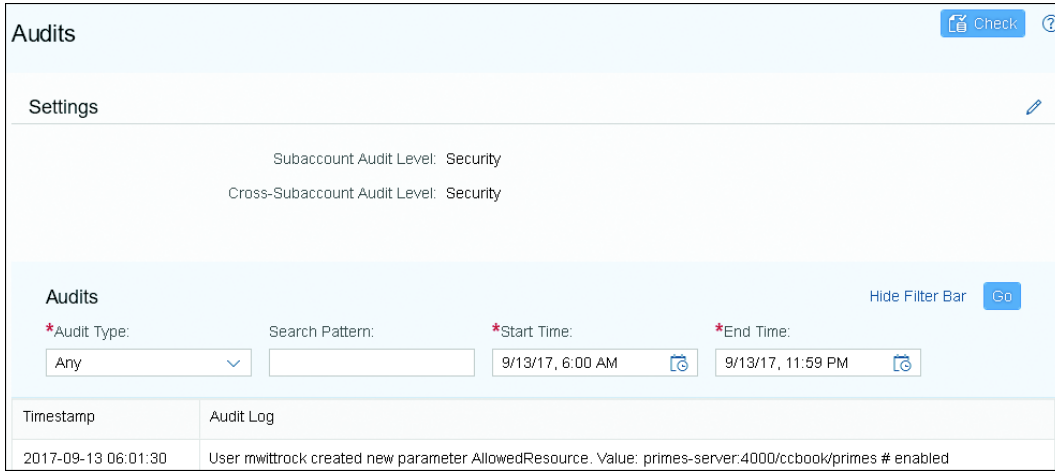


Figure 98 The Audits View in the Subaccount Menu

> **Note**
>
> The reason that the audit log is located in the subaccount menu is that it contains both entries that are specific to the current subaccount and entries that are independent of subaccounts. An example of the former is the entry written when a new backend resource is added. An example of the latter is the entry written when the authentication mode is changed.

The displayed audit log entries can be filtered using the filter bar, shown in Figure 99.

Figure 99 The Audit Log Viewer's Filter Bar

To limit the audit log list to only entries of a particular type, such as configuration change entries, select the desired type from the **Audit Type** dropdown list.

The list of log entries can also be limited to those that contain a particular search string, for instance, a user name. To do so, enter your search string into the **Search Pattern** field. To perform a wildcard search, use * to match from zero to any number of characters, and ? to match exactly one character.

You can also limit the list to those entries whose timestamps fall within a particular time interval. To do this, use the date and time controls of the **Start Time** and **End Time** fields to delineate the desired interval.

When you are happy with your filter settings, click the **Go** button to apply them to the list of audit log entries.

Audit Log Levels

The *log level* determines which entries Cloud Connector writes to the audit log. The current log levels for the subaccount and cross-subaccount auditing are displayed in the **Settings** section. The three available log levels are `Security`, `All`, and `Off`.

At log level `Security`, which is the default, Cloud Connector writes audit log entries for every significant configuration change performed by an administrator. Examples of such changes are activating payload trace, removing a backend resource, and changing LDAP settings. At this log level, Cloud Connector also writes a log entry for every backend resource request that was denied at runtime.

At log level `All`, in addition to the entries written at the `Security` log level, Cloud Connector also writes an audit log entry for every backend resource request that was allowed.

At log level `Off`, no entries are written to the audit log.

Note

SAP recommends always having audit logging activated at the `Security` log level in production scenarios. When necessitated by, for example, specific legal or industry requirements, the log level should be raised to `All`.

To edit the current log levels, click the **Edit** icon in the **Settings** section. This opens the **Edit Audit Level** dialog, shown in Figure 100.

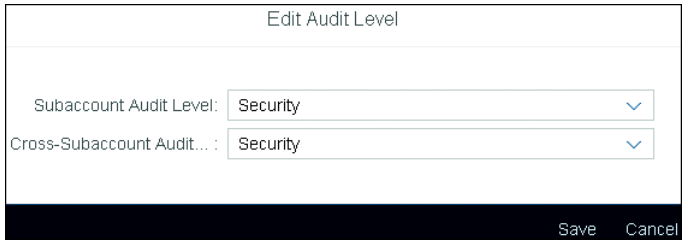


Figure 100 Editing the Audit Log Levels

To set the log level for the current subaccount, select the level from the **Subaccount Audit Level** drop-down list.

To set the log level for the cross-subaccount auditing, select the level from the **Cross-Subaccount Audit Level** drop-down list.

Then click **Save** to store the configuration and close the dialog.

Audit Log File Integrity and Security

The audit log files are stored in subdirectories of the *audit* directory, which is located in the *log* directory in the root of the Cloud Connector installation. SAP recommends that the audit log directories are backed up regularly and in accordance with local regulations.

Note

In a high availability setup, audit log files are written only on the master instance. This means that if a shadow instance takes over from an unavailable master instance, audit log files will be split between the two instances. For more information on high availability, please see Section 6.6.

The log files are cryptographically signed to make tampering detectable. You can perform a check of the integrity of the audit log files from within the Cloud Connector UI. To do so, click the **Check** button in the **Subaccount • Audits** view. This initiates a check of those log entries whose timestamps fall within the filter bar's current date and time interval.

If the integrity check detects an issue, you can get more details on the problem by running a command line script. The script is located in the *auditor* directory in the root of the Cloud Connector installation. On Microsoft Windows, it is a batch file called *go.cmd*. On both Linux and macOS, it is a shell script called *go.sh*.

Note

The scripts need to be able to execute the `java` command. Therefore, either your Java installation's *bin* directory must be added to the `PATH` environment variable, or the `JAVA_HOME` environment variable must be set, pointing to your Java installation directory.

In addition to this security measure, SAP recommends that Cloud Connector administrators are not also given administrator privileges at the operating system level. This ensures that a single individual cannot, for instance, alter Cloud Connector configuration and at the same time remove the evidence of having done so. Switching on your operating system's file access auditing will provide additional security by letting system administrators monitor for unauthorized changes to the audit log files at the file system level.

The Audit Log and Named Users

Whenever the action of an administrator causes an audit log entry to be written, the administrator's user name will appear in the text of the entry. This lets an auditor determine who performed a particular change. However, in password authentication mode, there is only one administrator user, regardless of the actual number of Cloud Connector administrators. This poses a problem: the audit log entry describes what was changed, but not by whom. As a consequence, the value of the audit log is greatly reduced in password authentication mode.

The solution to this problem is LDAP authentication. In this mode, every administrator has a separate, named user, and audit log entries therefore identify a specific person. This is one of the reasons LDAP authentication is the recommended authentication mode. For more information about LDAP authentication, please see Section 7.5.

7.8 The Trust Store

By default, Cloud Connector will establish a TLS connection to any backend system that will accept it. However, you can also configure Cloud Connector to allow TLS connections only to a list of trusted systems. This whitelist is known as the *trust store*, and it contains the public keys of the trusted backend systems.

Similar to the behavior of the trusted application whitelist, Cloud Connector trusts every backend system when the trust store is empty. However, once at least one public key has been added to the list, a TLS connection to an untrusted system will fail.

Surprisingly, an entry is not written to the audit log when a connection to an untrusted system fails. Instead, the details of the error must be found in the *ljs_trace.log* default trace file. For more information on log and trace files, please see Section 6.2.

To access the trust store, go to the **ON PREMISE** tab of the **Connector • Configuration** view. The **Trust Store** section, shown in Figure 101, lists the public keys of the trusted backend systems, along with the status of each key.

Figure 101 The Cloud Connector Trust Store

To add the public key of a trusted system, click the **Add a public key** icon +. This opens the **Add Public Key** dialog, shown in Figure 102.

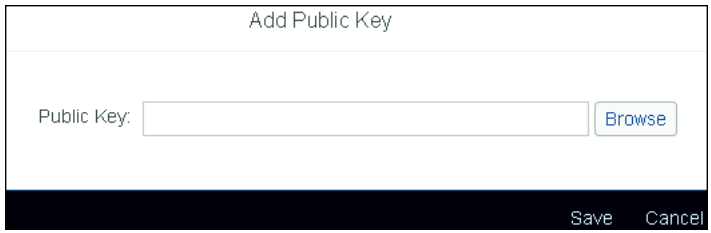


Figure 102 Adding the Public Key of a Trusted Backend System

Click the **Browse** button and select the key file from its location. Then click **Save** to add the public key to the trust store and close the dialog.

Note

Cloud Connector requires that the public key file's extension is either *.der* or *.cer*.

To delete a public key from the list, click its **Remove this public key** icon. To delete every public key from the list, click the list's **Remove all public keys** icon.

Acknowledgments

While writing the E-Bite you are now holding in your hands (figuratively speaking), I was very fortunate to have the help, support and encouragement of some great people.

I would like to thank Gregor Wolf, who was kind enough to spend time reading the draft manuscript, and providing some excellent, on-point feedback.

I also want to thank Mikael Krawczyk, who has been encouraging me to write for a long time.

I am grateful to Morten, Mikkel, Lars, Jan and Henrik for not kicking me out of our gaming group, when my writing started interfering with our regularly scheduled board game sessions.

At SAP, I would like to thank Udo Paltzer and Philipp Stehle for their support. A special shout-out goes to Matthieu Pelatan and Markus Tolksdorf, who tirelessly answered my many, many questions.

At KMD, I want to thank my colleague Allan Jakobsen for his insightful comments, and my manager Ole Jedrzejczyk for supporting the project.

At SAP PRESS, I would like to thank Florian Zimniak for believing in my idea, Hareem Shafi for her patience and support throughout the entire writing process, and the production crew for the awesome job they did, turning my manuscript into a great-looking E-Bite.

Last but most definitely not least, I want to thank my wife Betsey and our daughter Ada for their support, and for bearing with me while I was writing, which was pretty much all the time for a few months.

Every single person mentioned here helped make this work better. Any errors and omissions that might remain are entirely my own.

Morten Wittrock
September 2017

8 What's Next?

You've learned to set up and operate Cloud Connector to connect your on-premise systems with SAP Cloud Platform. Now you can begin fully integrating your cloud and on-premise landscapes using SAP Cloud Integration.

Recommendation from Our Editors

Learn to integrate cloud and on-premise landscapes with SAP Cloud Integration! Start with the basics, and then explore predefined integration patterns, debug and secure your integration projects, develop custom adapters, and more. Visit www.sap-press.com/3979 and check out *SAP HANA Cloud Integration*.

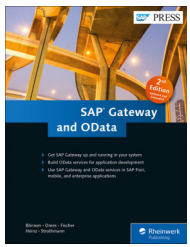
In addition to this book, our editors picked a few other SAP PRESS publications that you might also be interested in. Check out the next page to learn more!

More from SAP PRESS

Getting Started with SAP HANA Cloud Platform: SAP HANA Cloud Platform can do it all. With this book, get the basics of SAP HCP, and then take the next steps. Learn how to create, deploy, and secure applications. There's more to cloud than fluff—find out what it is.

519 pages, pub. 04/2015
E-book: $59.99 | **Print:** $69.95 | **Bundle:** $79.99
www.sap-press.com/3638

SAP Gateway and OData: Learn to create SAP Gateway and OData services, then expand your skills with how-tos on developing SAPUI5 apps with the SAP Web IDE; building SAP Fiori, mobile, and enterprise applications; and performing administrative tasks for lifecycle management and security.



785 pages, 2nd edition, pub. 04/2016
E-book: $69.99 | **Print:** $79.95 | **Bundle:** $89.99
www.sap-press.com/3904

SAP Process Orchestration: Learn to use the AEX to configure the System Landscape Directory, work with the ES repository, and manage the integration directory in SAP PO. Build integration flows, create a SAP BPM process, and get the most out of SAP BRM.

908 pages, 2nd edition, pub. 09/2017
E-book: $69.99 | **Print:** $79.95 | **Bundle:** $89.99
www.sap-press.com/4431

Usage, Service, and Legal Notes

Notes on Usage

This E-Bite is **protected by copyright**. By purchasing this E-Bite, you have agreed to accept and adhere to the copyrights. You are entitled to use this E-Bite for personal purposes. You may print and copy it, too, but also only for personal use. Sharing an electronic or printed copy with others, however, is not permitted, neither as a whole nor in parts. Of course, making them available on the Internet or in a company network is illegal.

For detailed and legally binding usage conditions, please refer to the section Legal Notes.

Service Pages

The following sections contain notes on how you can contact us.

Praise and Criticism

We hope that you enjoyed reading this E-Bite. If it met your expectations, please do recommend it. If you think there is room for improvement, please get in touch with the editor of the book: Hareem Shafi (hareems@rheinwerk-publishing.com).

We welcome every suggestion for improvement but, of course, also any praise! You can also share your reading experience via Twitter, Facebook, or email.

Supplements

If there are supplements available (sample code, exercise materials, lists, and so on), they will be provided in your online library and on the web catalog page for this book. You can directly navigate to this page using the following link: http://www.sap-press.com/4528. Should we learn about typos that alter the meaning or content errors, we will provide a list with corrections there, too.

Technical Issues

If you experience technical issues with your e-book or e-book account at SAP PRESS, please feel free to contact our reader service: support@rheinwerk-publishing.com.

About Us and Our Program

The website http://www.sap-press.com provides detailed and first-hand information on our current publishing program. Here, you can also easily order all of our books and e-books. Information on Rheinwerk Publishing Inc. and additional contact options can also be found at http://www.sap-press.com.

Legal Notes

This section contains the detailed and legally binding usage conditions for this E-Bite.

Copyright Note

Your Rights as a User

You are entitled to use this E-Bite for personal purposes only. In particular, you may print the E-Bite for personal use or copy it as long as you store this copy on a device that is solely and personally used by yourself. You are not entitled to any other usage or exploitation.

In particular, it is not permitted to forward electronic or printed copies to third parties. Furthermore, it is not permitted to distribute the E-Bite on the Internet, in intranets, or in any other way or make it available to third parties. Any public exhibition, other publication, or any reproduction of the E-Bite beyond personal use are expressly prohibited. The aforementioned does not only apply to the E-Bite in its entirety but also to parts thereof (e.g., charts, pictures, tables, sections of text). Copyright notes, brands, and other legal reservations as well as the digital watermark may not be removed from the E-Bite.

Digital Watermark

This E-Bite copy contains a **digital watermark**, a signature that indicates which person may use this copy. If you, dear reader, are not this person, you are violating the copyright. So please refrain from using this E-Bite and inform us about this violation. A brief email to info@rheinwerk-publishing.com is sufficient. Thank you!

Limitation of Liability

Regardless of the care that has been taken in creating texts, figures, and programs, neither the publisher nor the author, editor, or translator assume any legal responsibility or any liability for possible errors and their consequences.

Imprint

This E-Bite is a publication many contributed to, specifically:

Editor Hareem Shafi
Copyeditor Ruth Saavedra
Cover Design Graham Geary
Icon made by DinosoftLabs from www.flaticon.com
Layout Design Graham Geary
Production Marissa Fritz
Typesetting SatzPro, Krefeld (Germany)

ISBN 978-1-4932-1625-3

1st edition 2018